

(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(51) Int. Cl.
G06F 15/16

(11) 공개번호 특1999-0087923
(43) 공개일자 1999년12월27일

(21) 출원번호	10-1999-0013242
(22) 출원일자	1999년04월15일
(30) 우선권주장	9/072,677 1998년05월05일 미국(US)
(71) 출원인	인터넷서널 비즈니스 머신즈 코퍼레이션 포만 제프리 엘
	미국 10504 뉴욕주 마운트
(72) 발명자	헤이즈켄트필모어 주니어
	미국노스캐롤라이나27516체슬럼브룩크레인9309
(74) 대리인	원석희, 박해천

심사청구 : 있음

(54) 유저와그림의관계에서주제를선택함으로써관리자가원드유저매플리케이션을구성하는것을허용하는중
앙매플리케이션패니저먼트를가진클라이언트-서버시스템

요약

본 발명은 서버(server)와 복수의 유저 스테이션(user station)을 상호 연결하는 네트워크(network)를 가
진 시스템에 관한 것이다. 시스템은, 관리자(administrator)의 관계에서보다 유저(user), 또는 유저 그룹
(user group)의 관계에서 적절적으로 매플리케이션(application)을 작동시킴으로써 관리자가 유저 매플리
케이션을 구성하는 것을 허용하는 능력을 제공한다. 즉, 매플리케이션의 구성(configuration)이, 매플리
케이션을 실행하고, 그 목적을 위해 매플리케이션에 의해 제공되는 옵션(option)을 사용하는 매플리케이
션을 구성하고, 그리고 실제 유저 또는 그룹이 매플리케이션을 실행하는 것처럼 구성을 저장함으로써, 수
행된다.

도표

도5

백인어

클라이언트-서버 시스템; 어드미니스트레이터

발명자

도면의 간단한 설명

도 1은 본 발명이 실행되는, 관리자의 스테이션(station)을 포함하는 네트워크(network) 및 유저 스테이
션을 나타내는 도면.

도 2는 서버(server)와 통신하는 관리자 스테이션, 그리고 중앙 프로파일 매니저먼트(profile management)
및 퍼퍼런스 관리를 제공하는 관리자의 스테이션 및 서버의 컴포넌트(component)의 블록 다이어그램.

도 3은 시스템의 유저 그룹 및 유저의 계층적 조직을 나타내는 도면. 이 설명적인 계층적 조직은 개별적
터미널(terminal) 및 터미널 그룹(terminal group)을 포함할 수 있으나, 그러나, 이것들은 단순성을 위하
여 생략되었다.

도 4는 유저 그리고 유저에 의해 실행되는 특정 매플리케이션(application)에 적용하는 도 3의 계층적 조
직으로부터 퍼퍼런스의 셋(set)을 결정하기 위해 사용되는 개별 유저 및 그룹 우선 순위의 리스팅
(listing)을 나타내는 도면.

도 5는 도 2의 관리자의 스테이션 및 서버의 더욱 상세한 도면.

도 6은 유저의 터미널로서 매플리케이션의 실행 동안 유저 퍼퍼런스를 형성하기 위해 협동하는, 매플리
케이션과 다른 컴포넌트 사이의 유저 매플리케이션 그리고 API를 포함하는 유저의 터미널에서의 소프트웨어
객체의 설명적 도면.

도 7 내지 8은 유저 터미널에서, 데스크탑(desktop) 퍼퍼런스를 포함하는, 유저 로그-온(log-on) 그리
고 유저의 데스크탑을 처음으로 형성함을 위한 두 유저의 터미널 및 서버에서의 설명적 동작을 나타내는

도면.

도 9 내지 11은 두 관리자의 터미널 그리고 관리자 유지 로그-온을 위한 서버에서, 관리자의 데스크탑의 형성, 그리고, 예로서, 애플리케이션의 프래퍼런스 그리고 구성(configuration)을 위한 상황(context)을 위한 설명적 동작을 나타내는 도면; 이 예는 또한 구성 동안 상황 변경, 유지의 데스크탑 그리고 결과 동작을 나타낸다.

도 12 내지 24는 일 예의 표현인 도 3의 계층적 구조, 유지의 생성 및 삭제, 애플리케이션을 위한 애플리케이션 프래퍼런스의 형성, 그리고 프래퍼런스 형성 동안 상황 변경을 포함하는 애플리케이션 관리의 여러 측면에서 여러 형태의 실제 관리자 화면 스냅샷(snapshot)을 나타내는 도면.

* 도면의 주요 부분에 대한 설명

- 506 프로필 매니저(profile manager)
- 510 프로필 매니지먼트 프로퍼티즈 P(profile management properties P)
- 512 이벤트 리스너(event listener)
- 514 프로필 매니저 서블릿(profile manager servlet)
- 518 웹 서버(web server)
- 520 익스포트 에이전트(Export Agent)
- 108 관리자(Administrator)

발명의 상세한 설명

발명의 목적

발명이 속하는 기술분야 및 그 분야의 종래기술

본 발명은 일반적으로 퍼스날 컴퓨팅(personal computing) 및 네트워킹(networking)의 분야에 관한 것이다. 특히, 네트워킹 컴퓨팅의 새롭고 진화하는 분야에 관한 것으로, 여기서 데스크탑 컴퓨터 유지(desktop computer user)는, 통합 인트라넷(Intranet), 인터넷(Internet)과 같은 네트워킹에, 또는 네트워킹이나 인터넷 서비스 프로바이더(ISP)에 연결된, 가능한 디스크 없는(diskless), 퍼스날 컴퓨터를 나중 에 데스크탑 컴퓨터상에서 실행될 애플리케이션으로의 접속을 얻기 위해서 사용한다. 더욱이, 본 발명은 서버로부터 검색되고 데스크탑 컴퓨터에서 실행하는 위한 소프트웨어 프래퍼런스(preferance)(구성 데이터(configuration data))의 서버-기반(server-based) 저장에 관한 것이다.

네트워킹 컴퓨터의 분야는 현재 그것의 초기 단계에 있다. 그러나, 다수의 이유로, 특히 통합 환경에서, 급속히 진화하는 것이 예측된다. 예측 사항은, 회사 그리고 개인 유지들이 하드웨어 및 소프트웨어 업그레이드(upgrade) 시점에 도달하므로, 디스크 장착 컴퓨터 그리고 지역적으로 저장된 그리고 관리되는 소프트웨어 애플리케이션을 가진 종래 방법에서의 업그레이드보다 이 새로운 분야로 이행하는 것이 더욱 효과적이고 비용이 적게 든다. 예를 들면, 통합 환경에서 유지는, 예를 들면, 인터넷의 TCP/IP 그리고 HTTP 프로토콜(protocol)을 사용하여, 통합 인트라넷에서 연결할 수 있고, 소프트웨어 애플리케이션이 네트워킹 서버로부터 데스크탑 컴퓨터로 직접적으로 요구될 때 소프트웨어 애플리케이션을 다운로드할 수 있다. 애플리케이션이 유용한 작업을 수행하기 위하여 유지에 의해서 종래의 방법으로 데스크탑 상에서 실행된다. 이 구성(configuration)의 장점은 네트워킹 컴퓨터가 본질적으로 종래의 디스크 장착 컴퓨터보다 비싸지 않다는 것이다. 각 유지를 위해 소프트웨어의 개별적 카피(copy)를 구입하는 것보다, 유지를 위해 요구되는 다수의 소프트웨어 라이선스(licence)를 구매하는 것이 비용이 적게 든다. 확실히, 대다수의 통합 유지들을 수용하는 소프트웨어 관리 문제는 본질적으로 감소할 것이다. 현재, 디스크 장착 컴퓨터 또는 워크스테이션(workstation)의 각 유지는, 전문 기술의 부족으로 과도한 자원을 종종 소모하는 배경, 즉 실질적으로 그/그녀 자신의 시스템 관리자이다. 많은 유지들이 소프트웨어 설치, 업그레이드, 그리고 컴퓨터 관리의 문제와 싸우게 하는 것보다, 문제를 소수의 서버 관리 전문가에게로 효과적으로 떠넘김(offload)으로써 이 문제를 제거하는 것이 큰 장점이 될 것으로 예측된다.

전술된 바와 같이, 퍼스날 컴퓨터의 미래의 비전은 현재 그것의 초기 단계에 있다. 결과적으로, 존재하는 시스템에 많은 문제와 결점들이 있다.

전통적으로, 네트워킹 컴퓨터 시스템에서, 관리자는 네트워킹 서버 상에 저장된 유지 프로필(user profile)을 생성한다. 프로필은, 가령 유지 데스크탑 프래퍼런스 그리고 서버 상에 상주할 수 있는 다른 소프트웨어 애플리케이션으로의 접속을 위한 유지 허가과 같은 다른 형식의 정보를 포함할 수 있다. 유지가 시스템 상으로 로그 할 때, 유지는 서버로 그/그녀를 확인하고, 서버는 유지를 위해 프로필을 위치시키고, 그리고 서버가 컴퓨터를 구성하고 데스크탑을 발생하기 위해 사용되는 유지 컴퓨터로 프로필을 전송한다. 데스크탑은, 유지가 아마도 접속할 권한을 가지는 애플리케이션을 나타내는 다수의 아이콘(icon)을 포함할 수 있다. 프로필은 컴퓨터 및 데스크탑의 다른 애트리뷰트(attribute), 가령, 유지에 특정한 것이 데스크탑의 배경색, 또는 데스크탑 상에서 사용되는 캐릭터 폰트 그리고 포인트 크기, 또는 데이터 파일 검색 경로 등을 포함한다. 프로필은 유지 변경 가능 또는 비-변경 가능할 수 있다.

유지가 그를 자신의 프로필을 변경할 수 있는 환경에서, 변경된 프로필은 로그-오프(log-off) 시간에 서버로 다시 업로드되고(uploaded), 여기서 그것은 유지가 로그-온(log-on)하는 다음 시간에 검색하기 위해

저장된다. 어떤 종류의 기술에서, 우리가 가진 최상의 지식으로, 유저들이 서버 상에 존재하지 않는지, 유저가 서버 상에서 애플리케이션으로 실제로 접속 허가를 가지는지, 아니면 유저는 그룹의 데스크톱 상에서 그룹이 원하는 애플리케이션(아이콘의 어떠한 구성을 탐색할 수 있다. 로터스 워크플레이스 데스크톱(Lotus Workplace Desktop)(이전에는 Kona Desktop으로 불리었음) 시스템은 이 형식의 동작의 예이다. 다른 시스템에서, 서버는, 서버가 가지고 유저가 취할 수 있는 모든 애플리케이션의 유저로 리스트를 제공한다. 이 경우에, 유저가, 데스크톱 상에의 삽입(inclusion)을 위한 리스트로부터 선택되는 애플리케이션으로의 접속 허가를 가지는 것이 보장되지 않는다. 전 한 자바 뷰 시스템(Sun Hot Java Views system)이 이 시스템 형식의 예이다. 다시 말해서, 종래 기술의 시스템은 유저가 데스크톱 애플리케이션 아이콘의 셋트를 위해 구성할 수 있는 것과 유저가 실제로 접속 허가를 가지는 애플리케이션 사이에 상호 작용하지 않는다. 그러한 경우에, 유저가 애플리케이션을 수행하기 위해 아이콘을 클릭할 때, 접속허가가 존재하지 않는다면, 여러 메시지는 발생할 수 있고(가령 비인가된 접속 메시지), 또는 더욱 나쁜 경우에는, 유저의 컴퓨터는 일시 중단될 수 있다.

종래 기술이 가지는 다른 한계는 플랫 데이터 구조(flat data structure)가 모델 유저, 유저 그룹, 터미널, 그리고 터미널의 그룹으로 사용되는 것이다. 컴퓨터 자원으로의 유저 접속을 관리하기 위한 일반 계획 이후의 모델링된, 알려진 네트워크 컴퓨터 실시들(예를 들면, 데스크톱을 위한 로터스 관리 장비(Lotus Administration Facility for Desktops), 마이크로 소프트웨어 윈도우 NT 프로그램즈 앤드 폴리스시즈(Microsoft Windows NT Profiles and Policies), 전 한 자바 뷰즈(Sun Hot Java Views))은 여러 상황(context) 안에서 소프트웨어 프래퍼런스(또는 속성)을 관리하기 위해 서버 상의 플랫 그룹 구조(flat group structure)를 실시한다. 여기서 사용되는 바와 같이, "상황"(context)은 개별적 유저, 유저 그룹, 터미널, 또는 터미널 그룹을 말한다. 서버 상의 소프트웨어 프래퍼런스를 관리하기 위한 그룹핑 구조(grouping structure)는 관리자가 개별 유저들을 위한 것뿐만 아니라 유저들의 다른 그룹을 위한 프래퍼런스 속성을 정의하는 것을 허용한다. 그러나, 플랫 시스템(flat system)은 많은 환경에서, 특히 다수의 유저를 가지는 환경에서 유연하지 않다. 프래퍼런스 정보의 조직을 지원하는 관리 도구(administrative tool)를 계층적 구조(hierarchical structure)로 제공하는 것이 요구되고 있다.

종래의 시스템이 가지는 다른 한계는 그룹이, 관리자 또는 유저가 워크스테이션 데스크톱의 유저 구성을 수행해야 하는 방법에서 제한된다는 것이다. 예를 들면, 관리자들이 유저 애플리케이션과 떨어져 있으나 연관된 있는 구성 프로그램을 사용하여 유저 프래퍼런스를 구성하도록 현재 요구된다. 공급자(vendor)가 오직 단일 애플리케이션을 제공하는 것을 허용하는 것이 요구되고 있다. 공급자로부터 오직 엔드 유저 애플리케이션(end user application)을 요구하는 것은 중앙 매니지먼트 시설이 유저 또는 유저 그룹의 관계에서 엔드 유저 애플리케이션을 실행할 수 있는 것을 필요로 한다. 종래 기술은 동작의 이 관리 유연성을 허용하지 않는다. 다시 말해서, 종래 기술에서, 우리가 가진 최상의 지식으로, 관리자는 그 유저 및 애플리케이션을 위한 프래퍼런스를 셋트하기 위하여 유저의 관계에서 유저 애플리케이션을 작동시킬 능력을 가지고 있지 않다. 더욱이, 종래 기술에서, 관리자는 유저들의 그룹의 관계에서 프래퍼런스를 셋트하기 위하여 유저 애플리케이션을 작동시킬 수 있다.

발명자들에게 알려진 종래 기술의 또다른 한계는 종래 기술이, 서버 상의 다른 애플리케이션과 관련된 유저 프래퍼런스를 저장하기 위해 특정한 공간이 확보되는 것을 보장하기 위하여 서버 영구 저장 공간(server permanent storage space)을 분할하는 방법이다. 발명자들의 지식으로는, 객체가 그것을 다른 클래스들로부터 독특하게 확인하고 구별하게 하는 그것의 완전히 자격이 주어진 클래스 이름에 대해 정의될 수 있는 객체 지향 시스템에서 다른 애플리케이션을 위한 프래퍼런스 정보의 저장소 안에서 충돌을 방지하는 문제는, 제 1 중앙 권한이 공급자에 적용하는 독특한 지정을 할당하게 하고 그리고 공급자에 있는 제 2 권한이 각 공급자 애플리케이션을 위한 제 1 지정에 관련된 제 2 지정을 할당하게 함으로써 해결된다. 예를 들면, 공급자 A는 제 1 권한에 의하여 지정 공급자 A로 할당되고, 그 지정은 제 1 권한이 작동하는 구조 내에서 제 2 권한에 의하여, 지정 공급자 A에 할당되고, 그 지정은 그 구조 내에서 그것의 각 애플리케이션을 위한 제 2 지정을 할당한다. 예를 들면, 공급자 A의 애플리케이션들의 하나는 지정 공급자 A.App1로 지정될 수 있고, 다른 것은 공급자 A.App2로 지정될 수 있다. 종래 기술은, 다른 애플리케이션을 위한 프래퍼런스 데이터가 저장소 안에서 충돌하지 않는 것을 보장하기 위하여 시스템 안의 각 애플리케이션을 위한 독특한 지정을 시스템의 영구 저장소 안의 위치로 맵(mapping)한다. 애플리케이션은, 실행중일 때, 네트워크 컴퓨터 서버에게 그것의 독특한 저장 위치를 통보하고 그리고 다른 상황 안에서, 프래퍼런스 정보와 충돌하지 않도록 프래퍼런스 정보를 저장하기 위한 상황(유저, 유저 그룹, 터미널 또는 터미널 그룹)에 따라 시작 위치에서 구역을 분할하는 것은 서버의 책임이다. 명백히, 저장 공간을 관리하는 이 방법은 부적절하고 바람직하지 않다. 프래퍼런스 정보의 저장소 안에서 충돌을 방지할 목적으로 중앙 권한이 독특한 지정을 할당하게 하는 요구를 청하지 않고서, 그리고 저장 위치 정보를 애플리케이션으로 코딩하지 않고서, 전송된 객체 지향 애플리케이션을 위한 프래퍼런스 정보를 저장하기 위한 독특한 저장 위치를 자동적으로 할당하기 위한 방법을 고안하는 것이 요구된다.

종래 기술의 또다른 제한은, 존재하는 하드웨어 및 애플리케이션으로의 변화를 요구하지 않고서 존재하는 애플리케이션 및 하드웨어를 중앙 관리 네트워크 컴퓨팅 세계의 새로운 환경으로 이주하는(migrate) 준비의 부족에 있다. 네트워크화된(networked) 환경 안에서 예를 들면 터미널과 같은 존재하는 하드웨어는 서버 상에 위치한 특정 포트 안의 파일로부터 부트업(boot-up) 시간에 그것의 구성 정보를 취한다. 터미널은 그것의 구성 파일을 접속하기 위한 방법을 알기 위해 프로그램 된다. 터미널은 서버로부터 파일을 접속하기 위해 특정한 확인자(Identifier)를 사용한다. 특정한 확인자는 종종 터미널의 매체 접속 제어(media access control; MAC) 어드레스이다. 그러나, 프로토콜 및 터미널이 지정되는 것과는 다른 APIs를 포함하는 새로운 중앙 관리 환경 안에서, 터미널은 새로운 환경 안에서 프래퍼런스 정보를 접속할 수 없고, 터미널은 그것이 지정되는 방법에서 오직 그것의 구성 파일을 접속할 수 있다. 사용되고 있는 그러한 존재하는 디바이스가 많기 때문에, 이것은 심각한 문제이다. 새로운 시스템에서 그것들을 사용하는 무능력은 유저들이 새로운 시스템으로 이주하는 지극히 곤란적으로 제한한다.

종래 기술의 또다른 제한은 관리자 및 구성 매니지먼트 시스템 사이에서 인터페이스와 관련이 있다. 여러 유저 그리고 유저 그룹들, 그리고 터미널들 그리고 터미널 그룹들을 위한 프래퍼런스 정보를 구성하기 위한 관리 시설 내에서 소프트웨어를 구성할 때, 관리 소프트웨어는 시설을 작동시키고 있는 관리자에 의

하여 섯된 상황(유저, 유저 그룹, 터미널 또는 터미널 그룹)에서 시작한다. 관리자가 아래에서 애플리케이션이 움직이는 상황을 변경할 때, 애플리케이션은 새로운 상황을 위해 구성 정보를 로드하기 위해 재시작될 필요가 있다. 상황이 변경될 때마다 소프트웨어를 재 시작하는 프로세스는 특히 많은 유저를 가진 시스템에서 시간을 소모하고 관리자를 위해서 편리하지 않다. 그러한 시스템에서, 애플리케이션을 구성하는 동안 관리자가 여러 번 상황을 변경할 것이 예측된다.

발명이 이루고자 하는 기술적 과제

본 발명은, 관리자가 관리자의 관계에서 보다 유저(user) 또는 유저 그룹(user group)의 관계에서 애플리케이션(application)을 직접적으로 작동시킴으로써 유저 애플리케이션을 구성하는 것을 허용하는 능력을 제공한다. 즉, 애플리케이션의 구성(configuration)은 애플리케이션을 실행하고, 그 목적을 위해 애플리케이션에 의해 제공되는 옵션(option)을 사용하는 애플리케이션을 구성하고, 실제 유저 또는 그룹이 애플리케이션을 수행하는 것처럼 구성을 세이빙(save)함으로써 수행된다.

발명의 구성 및 작동

여기서 설명되는 시스템은 클라이언트-서버(client-server) 환경에서 유저 및 애플릿(applets)을 위해 구성(configuration) 정보를 위한 일반 저장소를 제공한다. 이것은 클라이언트 프로파일 매니저먼트(profile management)로서 언급된다. 시스템이, 유저가 돌아다니는 것(room), 즉 어떠한 컴퓨터로부터 시스템에서 어떠한 시간에 로그-인(log-in)하는 것 그리고 서버에서 유저를 위해 저장된 프러퍼런스에 따라 그것을 작동 시점에 자동적으로 구성되게 하는 것을 허용한다. 바람직한 실시예는 자바(JAVA는 Sun Inc.의 상표) 기반 시스템이고, 클라이언트 컴퓨터는 자바 애플리케이션(application)을 수행하기 위해 준비된 웹 브라우저 인터페이스(web browser interface)를 사용한다. 따라서, 바람직한 실시예에서, 유저 애플릿 및 데스크탑 애플릿은 자바 애플릿인 것으로 추정된다. 그러나, 본 발명을 자바 환경으로 제한하는 것으로 의도되는 것은 아니다. 서버-기반 애플릿을 위한 프러퍼런스가 여기서 설명되는 방법에서 다루어질 수 있는 반면에, 지역적으로 저장된 애플리케이션을 위한 프러퍼런스가 종래의 방법에서 지역적으로 저장될 수 있다.

바람직한 실시예에서, 시스템은 서버 및 복수의 유저 스테이션들을 상호 연결하는 네트워크를 포함한다. 서버는 유저로 다운로드 하기 위하여 복수의 유저 애플리케이션을 저장한다. 프로파일 매니저는 관리자 스테이션(administrator station)에서 제공된다. 프로파일 매니저는 제 1 엔드 유저 애플리케이션을 위하여 분리된 구성 애플리케이션을 실행하기 위하여 배치되고, 그것에 의하여 관리자는 시스템 유저의 다른 그룹 및 서버 그룹의 관계에서 제 1 엔드 유저 애플리케이션을 위한 구성 프러퍼런스를 특정할 수 있고, 그리고 서버 상에서 제 1 엔드 유저 애플리케이션을 위한 구성 프러퍼런스를 저장한다. 프로파일 매니저는, 다른 그룹 및 서버 그룹의 관계에서 제 2 엔드 유저 애플리케이션을 위한 구성 프러퍼런스를 특정하기 위하여 유저의 다른 그룹 및 서버 그룹의 관계에서 제 2 엔드 유저 애플리케이션을 수행하기 위하여 더 배치된다. 이것은, 프러퍼런스된 유저 또는 그룹 상황(context)에서의 관리자의 스테이션에서 제 2 엔드 유저 애플리케이션을 실행함으로써 그리고 서버 상에서 제 2 엔드 유저 애플리케이션을 위한 구성 프러퍼런스를 저장함으로써 행하여진다.

따라서, 본 발명은, 유저 또는 유저 그룹으로 행동하는 동안 효과적으로 엔드 유저 애플리케이션을 직접적으로 작동함으로써 관리자가 엔드 유저 애플리케이션을 구성하는 것을 허용한다. 본 발명의 부가된 장점은, 관리자가 유저 애플리케이션을 작동할 수 있고 그리고 유저가 보는 것과 동일한 화면을 본다는 것이다. 이것은, 관리자의 구성보다, 장소에서 유저 구성을 가진 유저의 관계에서 유저 애플리케이션을 작동함으로써 유저 문제를 진단하는데 있어서 관리자를 지원한다.

여기서 설명되는 시스템은 클라이언트-서버 환경 안의 모든 유저 및 애플릿을 위한 구성 정보를 위한 일반 저장소를 제공한다. 이것은 클라이언트 프로파일 매니저먼트로 언급된다. 시스템은, 유저가 돌아다니는 것, 즉 어떠한 시간에 시스템에서 어떠한 컴퓨터로부터 로그-인(log-in)하는 것 그리고 서버에서 저장된 프러퍼런스를 따라 작동 시점에 그것을 자동적으로 구성되게 하는 것을 허용한다. 바람직한 실시예는 자바(JAVA는 Sun Inc.의 상표임) 기반 시스템이고 그리고 클라이언트 컴퓨터는 자바 프로그램을 수행하기 위해 준비된 웹 브라우저 인터페이스를 사용한다.

용어 "애플릿"("applet") 및 "서블릿"("servlet")은 자바 프로그램 언어 기술에서 제정된 용어이고, 그 용어들이 이 분야의 숙련자들에게는 의미가 통하므로 여기서 사용될 것이다. "애플릿"("applet")은 자바 가능 웹 브라우저 내에서 작동하는 독립적인 소프트웨어 모듈을 언급한다. 서블릿은 자바 가능 웹 서버 상에 상주하는 소프트웨어 모듈을 언급한다. 용어 "애플릿"과 "서블릿"은 여기서 어떠한 방법으로라도 본 발명을 제한하는 것으로 이해되어서는 아니된다. 명확한 설명을 위해서, 구문 "구성 애플릿"("configuration applet")은 워드 프로세서, 데이터 베이스 매니저 등과 같은 엔드 유저 소프트웨어 애플리케이션을 위한 프러퍼런스를 구성하기 위해 사용되는 소프트웨어 모듈을 언급하기 위해 여기서 사용된다. 소프트웨어 애플리케이션이 역시 자바 환경 안에서 "애플릿"이므로, 구문 "유저 애플릿"("user applet") 또는 "애플릿"은 엔드 유저 애플리케이션을 언급하기 위해 여기서 사용된다.

바람직한 실시예에서, 유저 애플릿 및 데스크탑 애플릿은 자바 애플릿으로 가정된다. 그러나, 본 발명이 자바 환경으로 제한되지 않는 것으로 이해되어야 할 것이다. 본 발명은 어떠한 클라이언트-서버 시스템 안에서도 사용될 수 있다. 예를 들면, 요구되면, 시스템은, 어떠한 요구되는 프로그램 언어 안에서 기록되고 컴파일 되는(compiled) 특정의 통신 프로토콜 및 애플리케이션을 사용하기 위하여 설계될 수 있다. 더욱이, 바람직한 자바 기반 환경에서라도, 디스크 기반(disk-based) 컴퓨터는 지역적으로 어떠한 애플리케이션을 그리고 서버로부터 다른 애플릿을 접속할 수 있다. 서버-기반(server-based) 애플릿을 위한 프러퍼런스가 여기서 설명되는 방법에서 다루어질 수 있는 반면에, 지역적으로 저장된 애플리케이션을 위한 프러퍼런스가 종래의 방법에서 지역적으로 저장될 수 있다. 그러나, 바람직하게, 지역적으로 저장된 애플리케이션을 위한 프러퍼런스가 여기서 설명된 서버 기반 애플릿을 위한 프러퍼런스에 덧붙여 프로파일 매

니저먼트 프로퍼티즈 API(Profile Management Properties API)를 사용하는 서버 상에 저장된다.

단순한 애플리케이션 프로그램 인터페이스(Application Program Interface: API)는, 애플렛이 유저 또는 관리자에 의하여 실행될 때, API로 기록된 애플렛이 프러퍼티스 데이터를 용이하게 저장 및 검색하는 것을 허용한다. 애플렛 허가 및 유저 프러퍼티스는 그룹 멤버십(group membership) 및 개별 신원(individual identity) 상에 기반 하여 정의될 수 있다.

클라이언트 프로파일 관리는 아래의 서비스를 포함한다:

로그-온 지원 - 유저 프로파일로 맵핑(mapping);

유저 지원 - 유저 신분을 생성하고 유저로 서비스 및 프러퍼티스를 직접 제공하는 관리 능력;

유저 그룹 지원 - 유저의 계층적 그룹을 생성하고 그룹 멤버십에 기반한 서비스 및 프러퍼티스를 제공하는 관리 능력;

유저 애플렛 상황 투명성 - 유저 애플렛 실행의 상황의 자동적 결정. 즉, 유저 애플렛 실행 및 프로파일 환경의 자동 설정에 적용하는 유저 및/또는 그룹 프로파일의 결정;

유저 애플렛 프러퍼티스 저장소 - 유저 애플렛 구성 데이터를 위한 상황에 민감한(context-sensitive) 서버 저장소;

동적 유저 애플렛 프러퍼티스 인헤리탄스(inheritance) - 인헤리탄스의 객체 지향 원리를 통하여 유저 애플렛 프러퍼티스의 계층적 로드-타임(load-time)-합체; 및

유저 애플렛 접속 제어 - 그룹 디폴트 멤버십 특권에 기반한 유저 애플렛 실행의 제어. 관리자는 디폴트 그룹 특권을 무효화(override) 할 수 있고 개별 유저를 위한 부가의 접속 특권을 허가하거나 또는 거절할 수 있다.

프로파일 매니저먼트는 이 애플렛이 수행되는 프레임워크(iframework)를 제공한다. 어떤 일들은 프로파일 매니저먼트, 예를 들면 유저/그룹 매니저먼트, 애플렛 리스트, 상황 스위칭(context switching), 프러퍼티스 인헤리탄스 등에 의하여 직접적으로 지원되고, 반면에 유저 애플렛에 특정한 구성 서비스는 클라이언트 프로파일 매니저먼트 환경 내의 시스템 관리자에 의하여 분리되어 분리된 구성 애플렛에 의하여 일반적으로 지원된다. 어떤 엔드 유저 애플렛은 엔드 유저 애플렛의 부분으로서 구성 능력을 제공할 수 있다. 이 경우엔, 관리자는 그룹 유저 및 그룹을 위한 구성 프러퍼티스를 정하기 위해 개별 유저 및 그룹에 관련 하여 엔드 유저 애플렛(분리된 구성 애플렛에 반대하여)을 작동시킬 수 있다.

도 1은 본 발명을 실행하기 위하여 고안된 환경의 하나의 상위 조감도를 나타낸다. 네트워크(100)는 가령, 데스크탑 퍼스날 컴퓨터(102), 모바일 랩탑 컴퓨터(mobile laptop computer)(102), 워크스테이션(106)(예를 들면, RISC 컴퓨터), 관리자의 스테이션(108), 그리고 서버(110)와 같은 복수의 유저 스테이션을 상호 연결하기 위해 제공된다. 일 실시예에서, 네트워크(100)는 근거리 지역 통신망(local area network: LAN)일 수 있다. 다른 실시예에서, 네트워크(100)는 시스템 내에서 여전히 포함되는 사이트를 지리적으로 대신하는 회사와 같은 엔티티(entity)를 위한 광역 통신망(wide area network: WAN)을 포함할 수 있다. 본 발명이 실시되는 환경을 제한할 의도는 없으며, 실제로, 많은 형식의 스테이션을 상호 연결하는 어떠한 형식의 네트워크도 실현될 수 있다.

프로파일 매니저먼트 관리 인터페이스는 환경의 상위 다이어그램이 도 2에 나타내어진다. 관리자 클라이언트, 네트워크 컴퓨터(200)는 도 2의 왼쪽에 그리고 시스템을 위한 서버(202)는 도 2의 오른쪽에 나타내어진다. 클라이언트 및 서버는 (203)으로 나타내진 네트워크(203)를 통해 통신한다. 도 2의 특별한 예는 클라이언트 컴퓨터가 시스템 관리자의 컴퓨터임을 가정한다.

클라이언트 상의 프로파일 매니저(206)는, 관리자가 유저 및 그룹 레벨 둘 다에서 유저 애플렛 프러퍼티스를 구성하는 것을 허용한다. 관리자는 새로운 유저 및 그룹 계층을 생성할 수 있고, 유저를 다른 그룹으로 더할 수 있고, 각 그룹을 위하여 그리고 개별 유저를 위하여 애플렛 허가를 특정할 수 있다. 그리고, 관리자는 개별 유저 또는 그룹의 관계에서 애플렛을 구성할 수 있다. 관리자는 유저를 위한 패스워드를 더할 수 있고, 삭제할 수 있고, 리셋할 수 있다. 프로파일 관리 지원은 일반 유저에게 투명하다. 관리자는 어떠한 유저 또는 그룹의 관계에서 프로파일 매니저(206)를 부를 수 있다. 오직 관리자만이 클라이언트(유저) 및 그룹을 관리하기 위하여 그의/그녀의 상황으로부터 변경할 수 있다. 서버는 관리 권한이 없는 유저가 상황을 스위치 하는 것을 허용하지 않을 것이다. 요구가 서버로 올 때, 그것은 이 기능을 접속하기 위해 시도하는 유저의 인증된 ID를 필요할 것이다. 유저가 관리 권한을 소유하지 않는다면(즉, All Users Administrator group의 멤버가 아니면), 프로파일 매니저 서브넷(214)은 요구를 거절할 것이다.

프로파일 매니저(206)는 도 2에서 나타낸 바와 같은 애플렛(208)과 같은 다른 애플렛을 부른다. 이 예에서, 애플렛은 유저 데스크탑과 관련된 프러퍼티스를 구성하기 위하여 관리 애플렛일 수 있다. 또는, 애플렛은 편집기, 워드 프로세서, 데이터베이스 등과 같은 엔드 유저 애플렛과 관련된 구성 유틸리티일 수 있다. (208)와 같은 구성 애플렛들이 그룹의 상용하는 유저 애플렛으로부터 분리된 모듈로 존재하는 것이 요구되는 것은 아니지만, 바람직한 것이다. 도 2의 관계에서, 애플렛은 전형적으로 유저 애플렛을 위한 구성 애플렛이고, 그룹 프러퍼티스 및 허가 디폴트를 정하기 위해 관리자는 그룹 상황하에서 또는 유저 상황하에서, 개별을 위한 유저 애플렛 구성을 주문 제작하기 위하여 구성 애플렛, 애플렛을 작동시킨다. 애플렛은 그것의 유저 애플렛으로부터 분리된 모듈로서 실행함으로써, 구성 애플렛이 유저 애플렛과 비교되어 작을 수 있을 것이므로, 성능은 향상될 수 있다. 또한, 분리된 구성 애플렛은, 관리자가 유저 애플렛을 구성하기 위해 엔드 유저 능력을 제어하는 것을 허용한다.

전형적인 단독 운영(stand-alone) 컴퓨터는 그것의 유저 애플렛과 지역적으로 결합된 유저 애플렛 구성 정보를 저장한다. 전형적인 단독 운영 자바 기반 컴퓨터는 JavaUtil.Properties 클래스에 의해 제공되는 포맷을 사용하는 유저 애플렛 구성 정보를 저장한다. 둘 다의 배열은, 유저 애플렛이 유저 애플렛과 관련된 구성 정보를 저장하는 로컬 파일의 이름을 특정하는 것을 요구한다. 다시 말해서, 컴퓨터 및 그 위에 장착된 유저 애플렛 사이에 관계가 요구된다. 여기서 설명되는 프로파일 관리는 실제

java.util.Properties 객체의 익숙한 가능성을 제공하고, 이에 더하여 부가의 시설은 유저를 찾는(user-roaming) 가능성과 그리고 흠 없는(seamless) 플러그빌리티(pluggability)를 강력한 프레임워크(프로필 매니저)로 더 제공한다.

프로필 매니저먼트 프로퍼티즈 P(210)는 애플릿을 위한 프로퍼티즈 객체이고, 서버가 유저 및 그룹의 관제에서 애플릿을 위한 구성 정보를 어디에 저장할 것인가를 결정하는 것을 허용하는 서버 및 애플릿 1 사이에서의 API를 제공한다. 프로필 매니저먼트 프로퍼티즈 객체 클래스는 영구 저장으로부터 소프트웨어를 위한 구성 정보를 생성, 세팅, 그리고 검색을 제공하는 능력을 더 가지는 java.util.properties 클래스의 모든 기능성을 제공할 수 있다. 그러한 정보를 중앙 위치에 저장할 수 있는 유저 및 그룹 구성의 매니저먼트를 만든다. 유저가 관리자의 역할을 할 때, 프로필 매니저먼트 프로퍼티즈(210)는, 관리자 가 구성 애플릿에 대응하는 유저 애플릿을 구성하고, 또는 애플릿 1이 엔드 유저 애플릿이면 애플릿 1을 구성하고, 그리고 적절한 상황 안의 서버 상에서 적절한 위치 안에 구성 정보를 저장하는 것을 허용한다. 이것은 종래의 시스템에서처럼 유저 애플릿 및 컴퓨터 사이에서보다, 유저 애플릿 및 유저 사이에서 관계의 설립을 허용한다. 프로필 매니저먼트 프로퍼티즈(210)는 java.util.Properties 클래스의 확장이다. 이 확장은, 프로퍼티즈 객체 프로퍼티즈 정보의 키/값(key/value)의 쌍이, 스트림에 반대되고, java.util.Properties를 가진 키와 연관되는 것을 허용한다. 이것은, 교대로, 애플리케이션 디벨로퍼(developer)가 파일 이름과 경로보다도 프로퍼티즈 정보를 위한 상황과 관련된 특정한 위치를 특정하기 위해 키를 사용하는 것을 허용한다. 프로필 매니저먼트 프로퍼티즈(210)는 키를 자동적으로 결정한다. 키의 발생은 도 8 및 9와 연관되어 상술된다. java.util.Properties 클래스 다음에 프로필 매니저먼트 프로퍼티즈(210)를 모델링함으로써, 시스템은 순환적인 클래스-디플트 평가를 통해 프로퍼티즈 안헤리턴스를 활용할 수 있다. 따라서, 이 확장된 클래스는, 도 3에 관련하여 논의된 바와 같이, 현재 상황에서 시작하는 프로퍼티즈를 추적함으로써 그리고 디플트를 위한 전후 관계상의 계층을 트래버싱 업(traversing up)함으로써 "그룹 디플트"("group default") 능력을 제공한다.

서버(202)는, 유저 및 그룹 프로퍼티즈 그리고 유저 애플릿 접속 허가와 같은 유저 데이터 및 그룹 데이터를 저장하는 데이터베이스(212)를 포함한다. 웹 서버(218)는, 자바 애플릿을 위한 지원을 가진 전형적인 웹 서버를 나타낸다. 프로필 매니저 서버넷(214)은 유저 및 그룹 선언을 프로퍼티즈 데이터로 맵(map)한다. 그것은 서버 상에서 애플리케이션으로 유저 접속을 관리하기 위해 접속 제어 리스트를 또한 유지한다.

유저 및 그룹 프로퍼티즈는 도 3에서 나타낸 바와 같이 트리 구조(tree hierarchy)로 저장된다. 시스템의 모든 유저는 자동적으로 탑 그룹 AllUsers에 속한다. 모든 유저는 AllUsers 그룹에 속하고; 이 그룹은 서버 상에서 일부 또는 모든 유저 애플릿을 위하여 디플트 프로퍼티즈를 포함한다. 도 3에서, 서버는 App3, App4, App5로 선언 확인된 적어도 세 가지 유저 애플릿을 포함한다. AllUsers 그룹 안에서 지시된 바와 같이, App3를 위한 디플트 백그라운드(default background: BG)는 BG= 파란색(blue)이다. x, y 그리고 z로 명시된 다른 설명적인 프로퍼티즈는 상대적으로 1,2, 그리고 3의 디플트값을 가진 것이 나타내어진다. 용어 x, y, 그리고 z는 어떠한 요구되는 프로퍼티즈를 나타내기로 의도되고, 그리고 값 1,2, 그리고 3은 임의적이고 그리고 단지 그 점을 설명하기 위해 사용된다. x 프로퍼티즈는 예를 들면 데스크탑을 위한 스크린 폰트(screen font)일 수 있고; 값 x=1은 타임스-로만(Times-Roman)의 디플트 폰트를 요구할 수 있다. 유사하게, 모든 유저를 위한 App4를 위한 디플트 프로퍼티즈는 BG= 회색(gray), x=2, y=2, 그리고 z=20이다.

AllUsers 그룹 안의 디플트 값은 다른 유저 그룹 및 개별 유저를 위한 것과 같은 다른 상황을 위한 어떠한 요구되는 방법으로 변경될 수 있다. 예의 방법에 의해, 도 3에서 AllUsers의 상황에 덧붙여, 네 개의 그룹(그룹 X, 그룹 Y, 그룹 Y1, 그리고 그룹 Y2)이 나타내어진다. 덧붙여, 두 개별 유저1, 유저N이 나타내어진다. 유저는 한 개 그룹 이상의 멤버일 수 있다. 도 3에서, 유저1은 AllUsers, 그룹 X, 그룹 Y1의 멤버이고; 유저N은 AllUsers 및 그룹 Y2의 멤버이다. 유저가 한 그룹 이상의 멤버이면(AllUsers에 덧붙인 다른 그룹), 그룹은 그 유저를 위한 주어진 애플릿을 위해 프로퍼티즈를 고를 목적으로 우선 순위가 주어진다. 관리자는 유저를 위해 그룹 우선 순위를 구성한다. 그룹 우선 순위는 도 4에 설명된다. 도 4에서, 유저1은 그의/그녀의 가장 높은 우선 순위 그룹을 위해 그룹X(AllUsers, 그룹X의 완전히 자격 있는 이름으로 선언 확인됨)를 가진다. 유저1의 다음의 가장 높은 우선 순위 그룹은 그룹 Y1(AllUsers, 그룹 Y, 그룹 Y1)이다. 유저1의 가장 낮은 우선 순위 그룹은 AllUsers 그룹이다. 유저1이라고 하는 유저가 App3라고 하는 애플릿을 작동할 것을 요구할 때, 프로퍼티즈는 유저가 속하고 유저 애플릿이 유저 데스크탑 상에서 따라서 구성되는 그룹 또는 그룹들에 따라 도 3의 트리로부터 합쳐진다.

어떠한 상황을 위한 프로퍼티즈를 합쳐하는데 있어서의 제 1 스텝은 디플트를 취하는 것이다. 유저를 위한 디플트는, 만약 어떠한 것들이 있다면, 애플릿을 위한 프로퍼티즈 정보가 획득될 수 있는 가장 높은 우선 순위 그룹으로부터 애플릿을 위한 프로퍼티즈의 합쳐된 셋트이다. 그룹을 위한 디플트는, 만약 어떠한 것들이 있다면, 그룹 패어런트(즉, AllUsers 그룹은 AllUsers 그룹X의 패어런트(parent)이다)로부터 애플릿을 위한 프로퍼티즈의 합쳐된 셋트이다. 그룹이 패어런트를 가지고 있지 않다면(즉, 탑 레벨 AllUsers 그룹), 그 그룹을 위한 디플트는 없다. 상황에서 애플릿을 위한 프로퍼티즈를 합쳐하기 위해, 상황에 명백하게 저장된 애플릿을 위한 프로퍼티즈는 상황을 위한 애플릿을 위하여 디플트 프로퍼티즈를 오버라이트(overwrite)한다. 따라서, 그룹 상황 안의 애플릿을 위한 디플트 셋트로 프로퍼티즈를 합쳐하기 위해, 순환하는 콜(recursive call)은 각 그룹 노드로부터 애플릿을 위한 프로퍼티즈의 각 패어런트 셋트를 요구하는 AllUsers 그룹으로 만들어진다. 다음의 예를 설명하기 위해 도 3을 참조하기 바란다. 예를 들면, 상황이 AllUsers, 그룹 Y, 그룹 Y1이면, 콜은 애플릿을 위한 그것의 디플트 프로퍼티즈를 요구하는 그룹 Y인, 그룹 Y1의 패어런트로 만들어진다. 그룹 Y1은 순환적인 콜을, AllUsers인, 그것의 패어런트로 만든다. AllUsers는 패어런트를 가지지 않는다. 따라서 AllUsers는, 애플릿을 위해 그룹 Y로부터의 콜로 프로퍼티즈의 셋트를 리턴 한다. 이 프로퍼티즈의 셋트는, 어떤 경우에, 애플릿을 위해 그룹 Y에 저장된 프로퍼티즈에 의해 변경된다. 이것은 그룹Y1의 상황을 위해 애플릿을 위한 프로퍼티즈의 디플트 셋트이다. 디플트 프로퍼티즈의 이 셋트는 그룹 Y1으로부터 그룹Y로의 순환적인 콜의 결과로서 그룹 Y1로 리턴 된다. 그리고, 어떤 경우에, 이 예에서 사용되는 프로퍼티즈의 실제 셋트가 되기 위해 애플릿을 위한 그룹 Y1에서 프로퍼티즈에 의해 변경된다. 프로퍼티즈 정보가 유저를 위해서 획득될 수 있는 최

상위 우선 순위 그룹이 디폴트가 획득될 그룹 상황을 먼저 설립하기 위해 사용되는 것을 제외하고, 유저의 상황을 위한 프래퍼런스의 셋트가 동일한 방법 안에서 세워진다. 전송된 순환적인 프로시저는 유저 그리고 유저에 의해 요구되는 매들렛을 위하여 프래퍼런스의 실제적인 셋트를 세우기 위해 사용된다.

뒤따르는 예가 위의 프래퍼런스 합체를 설명하므로 도 3과 결합되어 읽혀져야 한다.

예 1 : 관리자는 그룹 AllUsers, 그룹 X를 위해 프래퍼런스를 셋트하기 위해 App3을 위한 구성 매들렛을 작동한다.

AllUsers, 그룹X의 관계에서 App3을 위한 프래퍼런스를 셋트하기 위하여, 프래퍼런스의 현재 셋트가 결정되어야 한다. AllUsers, 그룹 X는 그것의 패어런트 AllUsers를 위해 디폴트를 요구한다. AllUsers가 달 레벨 그룹이기 때문에, 그것은 그것의 프래퍼런스를 App3을 위해 Group X로 리턴 한다. 이것들은 Group X의 관계에서 App3을 위한 디폴트 프래퍼런스이다. Group X가 App3을 위해 프래퍼런스를 가지고 있지 않기 때문에, AllUsers로부터의 디폴트 셋트는 사용되어지는 프래퍼런스의 실제 셋트이다. 이 예에서, AllUsers 그룹으로부터의 이들 프래퍼런스는 : BG = 파란색(blue), x = 1, y = 2, z = 3. 관리자는, 어떠한 요구되는 방법에서 합체 프래퍼런스를 변경하기 위하여 구성 매들렛 사용을 지금 변경할 수 있다.

예 2 : 유저1은 com.ibm.App3의 실행을 요구한다. 프래퍼런스는, 유저1의 관계에서 com.ibm.App3을 위해 합체되어야 한다.

도 4는 유저1을 위한 최상의 우선 순위 그룹이 AllUsers, 그룹X임을 나타내고, 그룹 계층의 이 브랜치가 App3에 속하는 프래퍼런스 정보를 위해 먼저 검사될 것이다. 여기서부터, 프래퍼런스의 합체된 셋트가 유저의 워크스테이션 상에서 App3을 구성하기 위해 사용되는 것을 제외하고는, 이 예가 위 예1과 본질적으로 동일하다. App3을 위한 User1의 상황 안에서 저장된 BG= 녹색(Green) 프래퍼런스가 프래퍼런스 트리의 브랜치 AllUsers, 그룹X로부터 획득된 디폴트 BG= 파란색(Blue) 프래퍼런스를 무시하기 때문에, 유저1을 위한 App3을 위한 프래퍼런스는 : BG= 녹색(Green), x=1, y=2, z=3이다.

예 3 : User1의 관계에서 com.ibm.App6을 위한 합체 프래퍼런스.

이 예는 User1의 상황을 위한 합체되지 않은 프래퍼런스를 포함하는 최상위 우선 순위 그룹의 상태를 설명한다. 다시, 유저1을 위한 최상위 우선 순위는 그룹X이다. 이 그룹 및 그것의 패어런트 AllUsers는 App6을 위한 프래퍼런스를 포함하지 않는다. User1을 위한 다음의 최상위 우선 순위 그룹은 그룹Y이다. 프래퍼런스의 셋트가 App6을 위한 이 그룹으로부터 획득될 수 있다. 프래퍼런스의 합체가 예1에서 설명된 바와 같이 진행된다. 순환하는 물들은 그룹 Y1위의 트리로부터 루트(root) AllUsers 그룹으로 만들어지고, 프래퍼런스 셋트는 순환하는 물 아래로 리턴 되고, 디폴트 셋트를 구성하기 위한 방법을 따라 변경된다. 디폴트 셋트는 이 상황에 적응하는 프래퍼런스의 합체 셋트를 구성하기 위해 그룹 Y1 안에 저장된 프래퍼런스와 함께 변경된다. 간단히 설명되면, 그것이 App6을 위한 프래퍼런스를 가지고 있지 않기 때문에, AllUsers는 프래퍼런스의 널(null)셋트를 리턴 한다. 그룹 Y는 a=1 및 b=2 와 함께 이 널 셋트를 변경하고, 이 셋트를 디폴트 셋트로서 그룹Y1로 리턴 한다. 그룹 Y1은 a=33과 함께 디폴트 셋트를 변경한다. 이 셋트는 그것의 디폴트 셋트로서 사용을 위해 User1의 상황으로 리턴 된다. User1 상황에서 App6을 위한 프래퍼런스가 없기 때문에, 프래퍼런스 트리의 그룹 Y1 브랜치로부터 획득되는 디폴트는 App6을 위한 완전히 합체된 프래퍼런스 셋트를 나타낸다. 프래퍼런스의 실제 셋트는 이 상황을 위해 a=33, b=2가 된다.

위의 세가지 예는 소프트웨어 의 특정 부분을 위한 로드(load)()에 응답하여 프래퍼런스의 수정을 설명한다. 프래퍼런스 정보가 소프트웨어 부분을 위해 세이브될 때, 상황에서 명백히 기록된 세이브되는 프래퍼런스는 프래퍼런스가 저장된 소프트웨어를 위한 키이고 그리고 작동되는 상황 소프트웨어의 결합에 의해 지정되는 위치에서 데이터 저장(212)으로 기록될 것이다.

허가는 유사하게 동작한다. 새로운 그룹은 그것의 슈퍼 그룹에 의해 허가되는 모든 매들렛뿐만 아니라 그룹 그 자신에 의해 허가되는 모든 매들렛 이름으로의 접속을 가진다. 그러나, 자바는 프로그래머가 슈퍼 클래스 방법을 무시하는 것을 허용하는 것처럼, 프로그램 매니저먼트는 시스템 관리자 승력이 인헤리트(inherited)된 허가를 무시하는 것을 허용한다. 이것은 허가를 무시하는 것(overriding)으로 불린다.

인헤리턴스의 자바의 형태를 가지고서, 프래퍼런스 및 허가 인헤리턴스의 프로그램 매니저먼트 형태는 단일 인헤리턴스로 불린다. 단일 인헤리턴스는 각 프로그램매니저먼트 그룹이 오직 하나의 슈퍼 그룹을 가질 수 있는 것을 (어떠한 주어진 슈퍼그룹이 복수의 슈퍼그룹을 가질 수 있다하더라도)의미한다.

프로그램 매니저먼트 유저(leaf node)는 복수의 그룹에서 멤버십을 필요로 할 수 있다. 그룹으로써 시설은, 크로스 그룹 브랜치 합체에 의해서 소개되는 양될 수 없는 변하는 서브 셋트의 소개 때문에 변질된 구성의 변화를 최소화하기 위하여 프래퍼런스 인헤리턴스를 단일 계층 그룹으로 제한하는 것이 요구된다. 유저의 그룹 멤버십이 우선 순위로 지정되는 것을 허용함으로써, 특정의 매들렛과 관련된 프래퍼런스를 찾을 때 프로그램 매니저먼트는 검색 순서를 뒤따를 수 있다. 다시 말해서, 최상위 우선 순위를 가진 그룹과 함께 시작할 때 검색은 그것의 프래퍼런스를 로드 하려는 매들렛을 위해 구성 데이터를 포함하는 것으로 발견된 제1그룹에서 멈출 것이다.

유저는 그룹 멤버십으로부터 소프트웨어 허가를 불러받는다. 조심스러운 기업 모델링과 함께, 관리자는, 패들을 통하여 항해(navigate)하여야 할 필요 없이, 한번에 하나의 유저씩, 많은 유저로의 소프트웨어 접속을 지정할 수 있다. 프로그램 매니저먼트는 웹 서버가 매들렛으로의 접속을 허가/거절 하도록 프로그램 명령으로써 접속을 제어한다. 웹 서버는 접속제어를 실시한다. 프로그램 매니저 서버렛은 인증 목적으로 웹 서버로 공유되는 유저 ID(ID#) 및 패스워드를 요구하는 웹 서버에 의하여 보호된다. 요구 되는대로 유저 패스워드를 불러주는 것은 표준 브라우저(browser)기능이다.

도 5는 도 2의 시스템을 더욱 상세하게 나타낸다. 구성 매들렛 매들렛1은 프로그램 매니저먼트 프레임워크 내의 관리자에 의해 불러여진다. 매들렛1은, 프로그램 매니저먼트 프레임워크 내에서 밀접하게 접속하기 위해 그것의 동작환경(예를 들면, 집의 상황, 상황이 변경된 이벤트, 이 상황들을 위한 집의 접속 제어 리스트)에 관한 정보를 집의하기 위해 매들리케이션 프로그램 인터페이스(API)(515)를 실시하고, 이것은 구

성 애플릿을 위한 요구가 아니다. 어떠한 이벤트에서 애플릿의 설계자는 오직 기본적인 API 방법 : `java.util.Properties` 객체로 그리고 객체로부터 프래퍼티스정보를 얻기 위해 사용되는 `java.util.Properties` 객체의 기본적인 방법에 부가하여 인에이블 퍼시스턴스(`enable Persistence()`), 로드(), 그리고 세이브()를 이해할 필요가 있다. API(515)는 부가적으로 리스트() 그리고 `getContext()` 방법을 제공한다. 애플릿은 오직 프로파일 매니지먼트 프로퍼티즈 클래스와 함께 등록할 필요만 있고, 이 방법들을 적절한 것으로 호출한다. 로드() 방법은, 관리자에 의해 프래퍼티스되는 유저 또는 그룹의 관계에서 구성되는 유저 애플릿을 위해 프래퍼티스의 현재 상태를 검색하기 위해 호출될 수 있다. 관리자는, 요구되는 대로 프래퍼티스를 변경할 수 있고 그것의 프로파일 매니지먼트 프로퍼티즈 객체의 세이브() 방법을 사용하여 애플릿에 의해 제공되는 구성 세이브 가능성을 사용하는 그룹을 저장할 수 있다. 유사하게, 애플릿이 접속을 위해 유저에 의해 승인되는 유저 애플릿의 리스트를 필요로 한다면, 그것은 서버로부터 리스트를 획득하기 위하여 리스트() 방법을 사용할 수 있다. `getContext()` 방법은, 그것이 작동하고 있는 상황의 이름을 나타내기 위하여 또는 그것이 오직 어떠한 상황에서 작동하는 것을 보장하기 위하여 애플릿에 의하여 사용될 수 있다(즉, 애플릿이 익스포트 에이전트를 사용하는 서버 상에서 서비스를 구성하기를 원한다면, 익스포트 되는 구성이 특정의 유저에 반대되는 바와 같은 특정의 서버이기 때문에, 그것은 그 자신이 `SiteUsers`에서 작동되는 것을 오로지 허용할 수만 있다). 프로파일 매니지먼트 프로퍼티즈(410) 안에서 작동하는 애플릿에 대해, 요구되는 모든 것은 애플릿이 프로파일 매니지먼트 프로퍼티즈(410)와 함께 등록하고 프로파일 매니지먼트 프로퍼티즈 클래스의 실시 즉, `java.util.Properties` class의 확장이다.

프로파일 매니저 (506)는 또한 구성 애플릿을 위한 상황 변화 (API)(516)를 제공한다. 애플릿은 상황 변화 이벤트 리스너(listener)(512)를 실시할 수 있다. API(516) 및 이벤트 리스너(512)는, 관리자가 구성 애플릿 작동 동안 멈추거나 그것을 재 시작해야 할 필요 없이 상황을 변경하는 것을 허용한다. 예를 들면, 애플릿 유저 프래퍼티스를 구성할 때, 관리자는 구성동안 여러 번 상황을 변경할 것이다. 구성 애플릿이 그러한 이벤트로 리스너로서 등록된다면, 프로파일 매니저 (506)는 그것에게 API(516)를 통해 상황 변화를 통지할 것이다. 이것은, 애플릿이 각 새로운 상황을 위해 서버로부터 그것의 프래퍼티스를 재생(refresh)하는 것을 허용한다. 이벤트 리스너(API)없이, 새로운 상황을 위해 존재하는 프래퍼티스 정보를 참조하고, 그리고 프로파일 매니지먼트 애플릿에 의해 멈추어지고 재 시작하는 것을 방지하기 위해 선택되어진 후에, 애플릿은 관리자에 의해 종료되고 재 시작되어야 할 것이다. 등록하기 위해, 애플릿은 그것의 프로퍼티즈 객체 프로파일 매니지먼트 프로퍼티즈(510), 즉 그 자신을 등록할 `addContextChangeListener(API516)` 상에서 방법을 호출한다. 관리자가 새로운 상황을 정할 때, 프로파일 매니저 (506)는 객체(510)로 셋트 상황(API516)을 수행하고, 그것은 응답하여 이벤트 리스너(512)상에서 리로드(reload)방법(API516)을 호출한다. 이벤트 리스너(512)는 새로운 상황을 위해 서버로부터 새로운 프래퍼티스 데이터를 얻기 위해 그것의 프로퍼티즈 객체(510)로 로드 프로퍼티를 수행하고, 그리고 애플릿이 새로운 프래퍼티스 정보를 반영하기 위해 그것의 GUI 및 내부 변수를 경선하게 한다.

새로운 상황 안에서 구성 변화를 만들기 전에 로드()를 하려고 의도할 때, 위의 기능은 한 상황으로부터 네트워크 관리자 판독 데이터, 상황 변경, 그리고 세이브()와 함께 오버라이팅(overwriting)의 가능성을 포함한다.

관리자가 상황 변경을 강제할 때, 프로파일 매니저 애플릿에 의해 리스너가 정지되고, 파괴되고, 리로드되고, 그리고 재 시작될 것이기 때문에 애플릿은 등록하지 않는다.

프로파일 매니지먼트는 또한 이 프로파일 매니지먼트 환경으로 존재하는 하드웨어 및 소프트웨어의 용이한 리트로피팅(retrofitting)을 허용하기 위해 "프로퍼티즈 익스포트"서비스를 제공한다. 프로퍼티즈 익스포트 서비스는 프로파일 매니저 (514)가 유저, 그룹, 그리고 유저 애플리케이션뿐만 아니라 유저 워크스테이션(물리적 하드웨어)을 지원하도록 허용한다. 존재하는 워크스테이션이 프로파일 매니지먼트 프로퍼티즈(510)에 대해서 알지 못하기 때문에, 익스포트 서비스는, 워크스테이션 벤더가 프래퍼티스 정보를 세이브할 때 서버 상에서 특정 익스포트 에이전트(520)가 불리하지도록 특정하는 워크스테이션-구성 애플릿을 생성하는 것을 허용한다. 익스포트 태그(export tag)는, 워크스테이션이 구성될 때 의해 요구되는 어떠한 프로프리타리(proprietary)파일 포맷/파일 위치에서도 워크스테이션 구성 정보가 세이브되는 것을 지정하기 위하여 벤더-공급 클래스 (익스포트 에이전트(520)객체)의 예가 생성되고 익스포트 방법이 객체 상에서 불리하지는 것을 야기시킨다.

애플릿이, 현재의 프로파일 매니지먼트 시스템과 양립할 수 없는 존재하는 터미널을 위한 벤더에 의해 제공되는 구성 애플릿을 가질지, 벤더 또한 익스포트 에이전트(520)를 제공한다. 관리자는 프로파일 매니저 (506)를 작동시킴으로써 이 시스템에서 동작을 위한 터미널을 구성할 수 있고, 터미널이 구성되도록 상황을 정할 수 있고, 벤더 공급 구성 애플릿을 작동할 수 있고, 그리고 애플릿을 구성할 수 있다. 관리자가 구성을 세이브할 때, 서버로 전송되는 정보의 부분은 터미널이 구성되는 것을 확인하는 독특한 확인자이다. 전형적으로, 이것은 터미널의 MAC(Media Access Control)어드레스이다. 프로파일 매니저 서블릿(514)은 세이브 상에서 익스포트 에이전트가 저장되는 것을 검출한다. 프로파일 매니저 서블릿(514)은 익스포트 에이전트를 위한 요구를 지정하는 세이브된 프래퍼티스의 하나로부터 이것을 검출한다. 프래퍼티스는 아래의 키 값 쌍의 형태로 익스포트 태그를 지정한다.

```
xxxxEXPORT-AGENTxxxx={fully }
```

익스포트 에이전트의 익스포트 (Context context, config properties) 방법은 세이브 프래퍼티스 정보로부터 서버 상에서 하나 또는 그 이상의 파일 (522)을 생성하기 위해 프로파일 매니저 서블릿 (514)에 의하여 호출된다. 특정의 파일 또는 파일들이 애플릿으로부터 프로퍼티즈 정보에 부속되는 터미널의 독특한 확인자에 의하여 확인된다. 터미널이 나중에 부트 업(boot up)할 때, 터미널은, 프로파일 매니지먼트 시스템에 대해 독립적으로, 그것이 항상해왔던 동일한 방법으로 서버 상에서 파일(522)로부터 그것의 구성 정보를 위치하고 검색하기 위해 그것의 독특한 확인자를 사용한다.

도6은 클라이언트 컴퓨터 상에 작동하는 애플릿2를 설명한다. 애플릿2는 워드프로세서와 같은 엔드-유저 애플릿일 수 있다. 어떠한 이벤트에서, 애플릿2는, 그것이 요구하면, 도5의 (515)에 나타내어진 동일한 API 방법의 일부로 접속을 가진다. 애플릿2는 프래퍼티스를 검색하기 위해 로드 방법을 사용하고,

엔드 유저에 의해 변경될 수 있는 어떠한 프래퍼런스를 세이브하기 위해 세이브 방법을 사용한다. 관리자 와 관련 되어 전송된 바와 같이, 인메이플퍼시스턴스

(EnablePersistence)는 '유저와의 동일한 관계에서 매플렛2를 위한 프로파일 매니지먼트 프로퍼티즈를 초기 화하고, 서버 상에서 프래퍼런스 정보 저장을 확인하기 위한 특정한 키를 발생한다.

도7은 그의/그녀의 데스크탑을 설명하는 유저의 상황을 나타낸다. 클라이언트(700)상의 유저는 서버 상 에서 데스크탑 매플렛의 URL에서 그의/그녀의 웹 브라우저를 지시하고, 스텝(704)에서 메시지 (http://server/Desktop.html)를 보낸다. Desktop.html은 서버가 보호하는 파일이므로, 회담(challenge)은 클라이언트(706) 상에서 웹 브라우저로 재송신 된다. 클라이언트 상의 웹브라우저는 유저 ID 및 패스 워드를 위한 유저를 프롬프팅(prompting)함으로써 응답한다. 클라이언트는 유저 ID 및 패스워드 정보를 서버(708)로 송신한다. 유저 ID 및 패스워드는, 이 정보가 웹브라우저 자신에 의해 통과됨을 설명하기 위하여 도3의 (708)에 나타내어진다. 이 형식의 명칭은 동일한 것을 나타내기 위하여 다른 장소에서 사 용된다. 아마, 유저가 데스크탑 매플렛을 작동하기 위한 허가를 가지기 때문에, 요구는 승낙될 것이다 (honored).

데스크탑 매플렛을 위한 코드가 서버로부터 클라이언트로 로드되는 곳에, 클라이언트와 서버사이에 일련의 상호작용(나타내어지지 않음)이 있다. 데스크탑 객체는 생성되고 (712)에서 실행하기 시작한다. 데스크 탑 객체는 그것의 프래퍼런스 정보를(즉, 구성 정보를) 필요로 하고, 따라서 그것은, 그것을 부르는 엔드 유저를 위한 데스크탑을 맞추어 만들 수 있다(tailor). 결국, 데스크탑 객체의 초기화 프로세스의 부분 으로서, 데스크탑은 (714)에서 프로파일 매니지먼트 프로퍼티즈 객체 P를 생성하고, 이 객체는 데스크탑 매 플렛을 위해 서버로부터 유저의 프래퍼런스 정보의 복사를 로드하고, 가지고, 캐쉬하고, 셋트하고, 그리 고 세이브하기 위해 사용된다. 그리고, 데스크탑 객체는 API 콜(call) P:인메이플(enable) 퍼시스턴스 (데스크탑 객체(매플렛))를 수행하고(716), 프로파일 매니저 서블렛(214)의 URL와 함께 프로파일 매니지먼트 프로퍼티즈 객체 P를 초기화한다. 이 URL은, 이전에 서버로부터 로드된 데스크탑 매플렛의 URL로부터 유 도된다. 프로파일 매니지먼트 프로퍼티즈 객체 P는, 데스크탑 매플렛을 작동하는 유저를 위한 상황을 가지 기 위해, 요구(718)를 프로파일 매니저 서블렛(214)으로 송신한다. 이 경우에, 상황은 두 요소로 구성되며, 유저의 ID인 상황 이름, 그리고 이 경우에 유저의 상황 형식이다. 프로파일 매니저 서블렛은 요구(718)로 부터 유저 ID를 취하고, 유저 상황을 리턴 한다(719). (716)의 스텝(2)에서 프로파일 매니지먼트 프로퍼티 즈 객체 P는 데스크탑을 작동하는 유저의 상황과 함께 초기화된다. (716)의 스텝(3)에서, 프로파일 매니지 먼트 프로퍼티즈 객체 P는 그것의 완전한 자격 있는 클래스 이름을 위해 자바 데스크탑 객체 P를 필요의한 으로서 데스크탑 소프트웨어를 위한 특정한 키를 발생한다. 모든 자바 객체는 그들의 클래스 이름을 안 다. 이 특정한 키는, 데스크탑 매플리케이션을 위한 유저 특정 프래퍼런스 정보를 저장하기 위해 데이터 베이스(212)안에 특정한 위치를 특정하는 파라미터를 제공하기 위해 유저의 상황 정보와 결합된다. 어떤 한 요구되는 방법이 완전히 자격 있는 클래스 이름 그리고 유저 상황 정보로 구성되어 있는 스트링을 데 이터 저장 위치로 맵핑하기 위해 사용될 수 있다. 다음으로, 요구(720)는 데스크탑 매플렛을 위해, 유저 를 위해 맞추어진 프래퍼런스 정보를 취하기 위해 프로파일 매니저 서블렛(214)으로 송신된다. 상황 및 키 는 요구된 프래퍼런스 정보를 확인하기 위해 요구(720)의 부분으로서 통과된다. 프로파일 매니저 서블렛 (214)은 요구된 프래퍼런스 정보(722)에 응답하고, 이 정보는 프로파일 매니지먼트 프로퍼티즈 객체 P(604) 안에서 캐쉬(cache)된다.

도8에서 계속하여, 데스크탑 객체(600)는 그것의 프로파일 매니지먼트 프로퍼티즈 객체 P로부터 그것의 프 래퍼런스 정보를 판독하고, 따라서 데스크탑을 갱신함을 시작한다(즉, 그것은 화면색을 푸른색으로 정할 수 있고, 마이콘의 위치에 관한 정보 등을 취할 수 있다). 데스크탑 객체는, 유저가 접속 허가를 가지는 소프트웨어의 리스트를 취하기 위해 그것의 프로파일 매니지먼트 프로퍼티즈 객체 P 상에서 방법을 호출한 다. 프로파일 매니지먼트 프로퍼티즈 객체 P는 프로파일 매니저 서블렛(214)으로부터 정보를 요구하고(802), 이 서블렛은 요구된 정보(804)와 함께 응답을 발생한다. 유저가 접속을 가지는 각 그러한 매플렛에 대해, 정보는 유저의 친근한 이름, 매플렛의 URL, 매플렛을 위한 아이콘의 URL(데스크탑이 데스크탑 상에 서 매플렛을 나타내기 위하여 그리고 그것을 로드하고 학습하기 위해 요구되는 정보) 그리고 본 발명에 관련이 없는 다른 프래퍼런스적 요소(material)를 포함한다. 이 정보는 프로파일 매니지먼트 프로퍼티즈 객체 P 안에 저장되고, 데스크탑 객체로 리턴 된다. (806)에서, 데스크탑 객체는 매플렛을 위한 폴더 (folder)를 구축하기 위해, 그리고 유저가 접속을 가지는 각 매플렛을 위한 윈도우가 아이콘 및 유저의 친근한 이름을 디스플레이 하는 것을 발생하기 위해 매플렛 정보를 사용한다.

유저에 의한 데스크탑의 이전의 작동에서, 유저는 설명된 폴더에서 디스플레이된 어떤 소프트웨어의 일부 분을 위한 아이콘을 드래그(drag)하고 드롭(drop)한다. 유저가 이때에 폴더로부터 데스크탑으로 드래그 되고 드롭된 매플렛으로 더 이상 접속을 가지지 않는 것이 가능하다. 그러나, 이 데스크탑 객체는 일반 적으로 마지막 작동 동안 세이브된 유저 프래퍼런스의 부분일 수 있고, 데스크탑 상에서 여전히 디스플 레이될 수 있다. 이 상태를 피하기 위해, 데스크탑은 유저가 접속을 가지는 모든 매플렛을 디스플레이하 기 위해 발생되는 윈도우(window)로부터 벗어나 나타나기 위해 구성된 매플렛을 위해 점검하기 위하여 그 것의 프로파일 매니지먼트 프로퍼티즈 객체 P로부터 그것의 프래퍼런스를 검사한다. 도8은 발생하는 매플 렷 윈도우로부터 벗어나 오작 한나의 매플렛이 있다는 것을 가장한다. 매플렛 윈도우의 바깥에 그러한 매플렛이 하나이상 있다면, 뒤따르는 프로시저는 각각의 그러한 매플렛을 위해 루프(loop)될 것이다. 스 텝(810)에서, 데스크탑은, 유저가 접속을 가지는 서버로부터 매플렛의 리스트에 대하여, 매플렛 윈도우의 바깥에 나타나는 이러한 각각의 매플렛을 점검한다. 리스트 안에서 매플렛이 나타나면, 매플렛을 위한 아이콘이 이전과 같이 동일한 위치 안에서 데스크탑(810)상에 위치한다. 유저가 매플렛으로 접속을 더 이상 가지지 않는다면, 매플렛은 스텝(814)에서의 데스크탑 프래퍼런스로부터 제거되고, 프로파일 매니저 먼트 프로퍼티즈 객체 P로부터 제거된다. 어떤 매플렛이 이 프로세스의 부분으로 제거되면, 데스크탑은 프로파일 매니지먼트 프로퍼티즈 객체 P가 스텝(816)에서 프래퍼런스를 세이브하도록 말한다. 프로파일 매니 지먼트 프로퍼티즈 객체 P는 데이터베이스(212)안에서 새로운 프래퍼런스정보를 세이브하기 위하여 프래 퍼런스 키, 그리고 프로파일 매니저 서블렛(214)으로 상황 정보를 가진 요구(818)를 송신한다. 서버는 프로파일 매니지먼트 프로퍼티즈 객체 P에 요구가 성공적으로 완성되었음을 통지하는 프로파일 매니지먼트 프 로퍼티즈 객체 P로 응답(820)을 송신한다.

도 9는 다른 유저 또는 유저의 그룹을 위한 애플릿을 위해 프래퍼런스를 구성하기 위해 구성 애플릿을 작동하는 관리자의 상태를 설명한다. 여기서 논의된 원리가 터미널 또는 터미널의 그룹의 구성에 일반적으로 적용되는 것이 이해되어진다. 클라이언트(900)상의 관리자는 서버 상에서 프로필 매니저 애플릿(214)으로 그의/그녀의 웹 브라우저를 지시한다. URL은 서버로 송신된다(904). ProfileManager.html이 서버가 보호하는 파일이므로, 응답은 클라이언트 상에서 웹 브라우저로 재송신된다(906). 웹 브라우저는, 유저 ID 및 패스워드를 위한 관리자를 프로파일 함으로써 응답한다. Profile Manager.html을 얻기 위한 요구는 메시지 안에서 포함되는 유저 ID 및 패스워드와 함께 서버로 (908)에서 반복된다. 아마도, 관리자가 프로필 매니저를 작동할 허가를 가진다면, 요구는 승낙되고 프로필 매니저 애플릿은 관리자 터미널(910)로 다운로드된다. 프로필 매니저 애플릿을 위한 코드가 서버로부터 클라이언트로 로드되는 클라이언트와 서버 사이에 (여기에서는 나타내어지지 않음)일련의 상호작용이 있다. 프로필 매니저 객체는 생성되고 스텝(912)에서 실행하기 시작한다.

프로파일 매니저먼트 프로퍼티즈 비 상황 플로팅(non Context Floating)은 일반적인 프로파일 매니저먼트 프로퍼티즈 객체 대신에 프로파일 매니저에 의해 사용된다. 그것은 하나의 예외와 함께 프로파일 매니저먼트 프로퍼티즈 객체로서 동일한 행동을 가진다: 프래퍼런스들이 로드되고 세이브될 때 그들은, 관리자가 구성하는 상황(즉 유저 또는 유저 그룹)으로 그리고 상황으로부터 로드하고 세이브하는 것에 반대하여, 프로파일 매니저를 작동하는 관리자의 상황으로 그리고 상황로부터 로드되고 세이브된다.

프로파일 매니저 객체는 그것의 프래퍼런스 정보(즉, 구성 정보)를 필요로 하고, 따라서 관리자가 그것을 부르는 프로파일 매니저를 맞춤 제작(tailor)할 수 있다. 결국, 프로파일 매니저 객체의 초기와 프로세스의 부분으로써, 프로파일 매니저는 스텝(914)에서 프로파일 매니저먼트 프로퍼티즈 비 상황 플로팅 객체 P.NCF를 생성하고, 이 객체는 프로파일 매니저 애플릿을 위해 서버로부터 관리자의 프래퍼런스 정보의 카피를 로드하고, 취하고, 캐쉬하고, 정하고, 그리고 세이브하기 위해 사용된다. 프로파일 매니저 객체는 P.NCF.enablePersistence(프로파일 매니저 객체 (애플릿))를 호출하고, 이 객체는 스텝 (916의1)에서 프로파일 매니저 서블렛(214)의 URL과 함께 프로파일 매니저먼트 프로퍼티즈 비 상황 플로팅 객체 P.NCF를 초기화한다. 이 URL은 프로파일 매니저 애플릿의 URL로부터 유도된다. 프로파일 매니저먼트 프로퍼티즈 비 상황 플로팅 객체 P.NCF는 관리자 및 상황 형식(USER)의 상황 이름(ID)을 얻기 위해 요구(918)를 프로파일 매니저 서블렛(214)으로 송신한다. 프로파일 매니저 서블렛은 요구(918)로부터 관리자의 ID를 얻는다. 웹 브라우저는 메시지 안의 관리자 ID 및 패스워드를 프로파일 매니저먼트 프로퍼티즈 비 상황 플로팅 객체 P.NCF에 의해 송신된 정보를 따라 통과시킨다. 프로파일 매니저먼트 프로퍼티즈 비 상황 플로팅 객체 P.NCF는 스텝(916의 2)에서 애플릿을 작동하는 관리자의 상황과 함께 초기화된다. 스텝(916의 3)에서, 프로파일 매니저먼트 프로퍼티즈 비 상황 플로팅 객체 P.NCF는, 그것의 완전한 자격이 주어진 클래스 이름(즉, Profile ManagerObject.setClass0.setName0)을 위하여 자바 프로파일 매니저 오브젝트 객체 (JavaProfile ManagerObject)(enablePersistence call 안에서 파라미터로 통과됨)를 정의함으로써 프로파일 매니저 애플릿을 위해 독특한 키를 발생한다. 관리자의 상황 정보와 결합된 독특한 키는, 프로파일 매니저 애플릿을 위한 관리자의 특정 프래퍼런스 정보를 위한 데이터베이스(212) 안에서 독특한 위치를 특정하기 위하여 업로드된다.

요구(922)는, 관리자를 위해 구성된 바와 같이 프로파일 매니저를 위해 맞춤 제작된 프래퍼런스 정보를 얻기 위하여 프로파일 매니저 서블렛(214)으로 송신된다. 요구(922)는 적절한 프래퍼런스 정보를 확인하기 위하여 적절한 상황 이름 그리고 형식 그리고 키 정보를 포함한다. 프로파일 매니저 서블렛(214)은 요구된 프래퍼런스 정보(924)에 응답하고, 이 정보 프로파일 매니저먼트 프로퍼티즈 비 상황 플로팅 객체 P.NCF 안에 캐쉬(cache)된다. 프로파일 매니저는 프로파일 매니저먼트 프로퍼티즈 비 상황 플로팅으로부터 그것의 프래퍼런스 정보를 판독하고, 따라서 그 자신을 갱신한다(즉, 예를 들면, 바탕색을 푸른색으로 정함).

동작은 도 10에서 계속된다. 프로파일 매니저는 프로파일 매니저 서블렛(214)으로부터 존재하는 유저들, 유저 그룹들, 그리고 소프트웨어에 대한 정보를 요구하고, 스텝(1002)에서 프로파일 매니저 구성 윈도우의 왼쪽 패널 안의 트리를 짓는다. 관리자의 왼쪽 패널의 예를 위해 도 13에서 24까지 참조 바란다. 이 시점(1004)에서, 관리자는 왼쪽 패널 트리로부터 유저 또는 그룹을 클릭함으로써 구조를 위한 요구되는 상황을 프래퍼런스한다. 프로파일 매니저는 P.NCF.setContext(선택된 상황)를 호출함으로써 프로파일 매니저먼트 프로퍼티즈 객체를 위한 상황을 정한다. "유저 그룹"의 선택된 상황을 위해 도 13을 참조하면, 이것은 모든 시스템 유저들의 그룹을 언급하며, 또는 도 18을 참조하면, 여기서는 "발명"("Development")의 그룹 상황이 선택되고, 또는 도 21을 참조하면, 여기서는 유저 상황 "콜랜드"("collend")가 선택된다. 다음에, 프래퍼런스(1006)에서, 관리자는 서버 상에서 모든 애플릿의 리스트로부터 구성되기 위해 애플릿을 선택한다. 애플릿 프래퍼런스의 예를 위해 도 17을 참조 바란다. 스텝(1008)에서, 관리자는 구성을 위하여 선택된 애플릿을 작동하기 위하여 작동/맞춤 제작(Run/Customize) 버튼을(button)을 클릭한다. 이 애플릿은 엔드 유저 애플릿을 위해 분리된 구성 애플릿일 수 있고, 또는 그것은 엔드 유저 애플릿 그 자신 일 수 있다. 선택된 애플릿은 서버(1009, 및1011)로부터 요구되고 로드된다. 스텝(1010)에서, 구성 애플릿 객체는 생성되고 그리고 실행하여 그것의 프로파일 매니저먼트 프로퍼티즈 객체 P를 발생하기 시작한다.

애플릿이 엔드유저 애플릿을 위한 분리된 구성 애플릿임이 가정되면, 스텝(1012)에서, 애플릿은 p.enablePersistence(ConfigApplet Object, fullyQualified ClassNameOfAppletBeingConfigured)를 호출한다. 반면에, 애플릿이 분리된 구성 애플릿보다도 유저 애플릿이라면, 콜미, 다른 애플릿을 위한 프래퍼런스정보에 반대되어, 그것 자신의 프래퍼런스 정보를 구성하기 원하기 때문에, 콜미는 p.enablePersistence(엔드 유저 애플릿 객체)일 수 있다. 현재의 상황은 관리자의 프로파일 매니저먼트 프로퍼티즈 비 상황 플로팅 객체 P.NCF를 경유하여 관리자에 의하여 이전에 셋트 되었기 때문에, 현재의 상황은 프로파일 매니저먼트 프로퍼티즈 객체 P에 의하여 이미 알려졌다. 가능(enable) 퍼시스턴스가 프로파일 매니저즈 프로파일 매니저먼트 프로퍼티즈 비 상황 플로팅 객체 P.NCF 상에서 호출될 때, 프로파일 매니저 서블렛(214)의 위치는 이전에 발생되었다. 구성 애플릿의 경우에, 애플릿을 위한 독특한 키는 가능 퍼시스턴스 콜 안에서 프로파일 매니저먼트 프로퍼티즈 객체 P로 구성 애플릿에 의하여 통과되기 때문에, 애플릿을 위한 독특한 키는 발생될 필요가 없다.

스텝(1014)에서, 구성 애플릿은 상황 변경 리스너로서 그것의 프로파일 매니저먼트 프로퍼티즈 객체와 함께

그 자신을 등록한다. 전술된 바와 같이, 이것은, 애플렛의 프로파일 매니저먼트 프로퍼티 객체가, 애플렛이 새로운 상황 안에서 종료되고 재학수되는 것을 요구하지 않고, 애플렛이 새로운 상황을 위해 프래퍼런스 정보를 로드할 수 있고 그리고 새로운 구성 정보를 반영하기 위해 그것의 그래픽 유저 인터페이스를 갱신할 수 있도록 하기 위해, 관리자가 상황

변경을 만드는지를 통지하는 것을 허용한다.

동작은 도 11에서 계속된다. 스텝(1104)에서, 구성 애플렛은, 프로파일 매니저먼트 프로퍼티 객체 P가 애플렛이 구성되는 것을 위한 현재의 상황로부터 프래퍼런스를 로드하도록 말한다. 요구(1105)는 애플렛이 구성되는 것을 위하여 관리자에 의해 이전에 선택된 상황을 위해 맞출 제작된 프래퍼런스 정보를 얻기 위하여 프로파일 매니저 서블렛(214)으로 송신된다. 요구(1105)는 적절한 프래퍼런스 정보의 위치를 지정 하기 위하여 적절한 상황 이름(상황 관리자가 선택한) 그리고 상황 형식(적절한 것으로 USER, USER GROUP, 또는 ALL_USERS_GROUP), 그리고 키 정보를 포함한다. 프로파일 매니저 서블렛(214)은 프로파일 매니저먼트 프로퍼티 객체 P 안에 캐쉬된 스텝(1106)에서의 요구된 프래퍼런스 정보에 응답한다. 구성은 프로파일 매니저먼트 프로퍼티 객체 P로부터 프래퍼런스를 얻고, 따라서 그것의 그래픽 유저 인터페이스를 갱신한다.

관리자는, 스텝(1107)에서, 예를 들면, 애플렛에 의해 제공되는 SAVE 버튼을 클릭함으로써, 애플렛을 구성하고 변경된 프래퍼런스를 세이브한다. 이 동작의 결과로써, 구성 애플렛은 그것의 프로파일 매니저먼트 객체 P 상에서 SAVE 방법을 호출한다. 프로파일 매니저먼트 프로퍼티 객체 P는 애플렛이 구성되는 것을 위한 프래퍼런스 그리고 특정한 키를 그리고 현재의 상황을 지정하는 정보를 프로파일 매니저 서블렛(214)으로 송신한다. 프로파일 매니저 서블렛은 상황 및 키에 의해 특정되는 위치 안의 데이터베이스(212) 안에서 프래퍼런스 정보를 저장한다.

구성 애플렛이 여전히 작동하는 동안, 스텝(1108)은 관리자가 지금 상황을 변경하는 예이다. 관리자는 유저 또는 유저 그룹을 클릭함으로써 새로운 상황을 선택한다(왼편 화면 패널의 관리자 안에서 새로운 상황을 위한 예를 위해 도 18을 참조). 상황 변경의 결과로써, 프로파일 매니저(506)는 P.NCF.setSelectedNewContext를 호출함으로써 셋트 상황 메시지를 프로파일 매니저먼트 프로퍼티 객체 P(510)로 송신하고, P.NCF.setSelectedNewContext는 객체 P가 이벤트 리스너(512)에게 리로드(reload) 프로퍼티 API(515)를 경유하여 상황 변경을 통지하도록 야기시킨다. 이것은 스텝(1110)에서 발생한다. 스텝(1112)에서, 이벤트 리스너(512)는 새로운 상황을 위한 프래퍼런스를 검색하기 위해 로드를 수행하고, 객체 P는 스텝(1118)에서 새로운 프래퍼런스와 함께 갱신된다. 관리자는, 요구되면, 새로운 상황을 위한 새로운 프래퍼런스를 갱신하고, 요구되면, 그것들을 세이브하도록 진행하고, 다음으로 전술한 바와 같이 필요하면, 새로운 상황 변경을 하면서 계속 진행한다.

나머지 도 12에서 24는, 프로파일 매니저(206)의 포션을 작동시키는 동안 관리자 워크스테이션의 실제 화면 스냅샷(snapshot)을 나타낸다.

주요 구성 윈도우(1200)가 도 12에 나타내어진다. 윈도우의 왼편에 있는 트리 뷰 패널(1202)은 프로파일 매니저먼트(1204)를 서버 상에서 활용 가능한 몇몇 서비스 중의 하나로 설명한다. 이 아이템(1204)이 도 12에 나타낸 바와 같이 선택될 때, 주요 윈도우의 오른 쪽 패널(1205)은 프로파일 매니저먼트 서비스를 위한 환영(welcome) 메시지를 디스플레이한다. 익스팬드(expand) 및 콘트랙트(contract) 아이콘(1208)은, 어떠한 것이 존재하면, 왼쪽 패널 안의 아이템의 출현(appearance)을 제어하기 위해 사용된다. (1208)은 "익스팬드 아이콘"으로 불리고, "프로파일 매니저먼트"("Profile Management") 아래에 서브 아이템이 있음을 지시한다. 관리자는 익스팬드 아이콘(1208)을 클릭함으로써 이 서브 아이템을 디스플레이 할 수 있고, 이것은 "콘트랙트 아이콘"("contract icon")(" ")이 할 것이다.

도 13은 도 12에서 프로파일 매니저먼트 아이템(1208)의 확장을 설명하고, 이 아이템은 도 13에서 세가지 디폴트 서브 아이템, 즉 "애플렛스"(1300), "유저 그룹"(1302), 그리고 유저(1304)의 디스플레이의 결과를 보인다. 익스팬드 아이콘은 이러한 아이템이 또한 확장할 수 있음을 지시한다. "애플렛"(1300)은 관리자가 서버(202) 상에서 활용 가능한 유저 애플렛을 정의하는 것을 허용하고, "유저 그룹"(1302)은 관리자가 도 3의 유저 그룹 트리를 생성하고 상주시키고, 그리고 그룹 프래퍼런스를 정하는 것을 허용한다. "유저"(1304)는 관리자가 새로운 유저들을 생성하고 그들의 프래퍼런스를 정하고 또는 존재하는 유저들을 위한 프래퍼런스를 변경하는 것을 허용한다. 도 13의 예에서, "애플렛"(1300)은 선택된다. 이 아이템이 선택될 때, 윈도우의 오른편에 있는 패널(1305)은 시스템으로 이미 정의된 유저 애플렛의 리스트(1306)를 디스플레이한다. (1306)에서 선택된 애플리케이션의 애틀리뷰트가 (1308)에서 나타내어진다. 관리자는 (1306)에서 <NEW>를 선택하고 (1308)에서 요구되는 이름 및 위치 정보를 입력함으로써 새로운 애플렛을 정의한다. 존재하는 애플렛 데이터베이스 익스플로러(Database Explorer)는 (1306)안에서 선택되어 나타내어진다. (1308)에서, "애플렛 이름"("Applet name") 필드는 이 애플렛 이름을 디스플레이한다. "URL"(Universal Resources Locator) 필드는 서버(202) 상에서 이 애플렛의 인터넷 또는 인터넷 웹 어드레스를 디스플레이한다. 필드 "html 파일의 완전한 경로"("Complete path of html file)는 서버(202)의 디스크 디렉토리 구조 안에서 애플렛의 디렉토리 경로 및 파일 이름을 디스플레이한다. 필드 "완전히 자격 있는 클래스 이름"("Fully qualified class name")은 애플렛의 완전히 자격 있는 클래스 이름을 디스플레이한다. 필드 "아이콘 URL"("Icon URL")은 유저 데스크톱 상에서 애플렛을 위해 아이콘을 발생하기 위해 사용되는 이미지 파일의 웹 어드레스를 디스플레이한다. 남아 있는 필드는 호출 즉시 소프트웨어에 의해 요구될 수 있는 선택적 정보를 위한 것이다. 코멘트 버튼(1310)은, "파일로부터 임포트 애플렛 리스트"("Import Applet List from File")은 관리자가 존재하는 텍스트 파일로부터 존재하는 리스트(1306)로 애플렛의 정의를 덧붙이는 것을 허용한다. 버튼(1310)이 클릭될 때, 도 14에서 나타내진 윈도우는 팝업(pop-up)하고, 관리자가 부가되는 애플렛 정의를 포함하는 텍스트 파일의 경로 및 파일 이름을 입력하는 것을 허용한다. 모든 보류 변화를 세이브하기 위하여, 관리자는 파일(1312)에서 클릭하고 세이브한다(나타내어지지 않음).

왼쪽 패널에서, 유저 그룹 아이템(1302)은 도 3의 AllUsers의 그룹에 응답한다("유저 그룹" 및 "AllUsers"는 여기서 호환되어 사용됨). "유저 그룹" 아이템(1302)이 선택될 때, 도 15는 관리자 스테이션의 오른쪽 패널을 나타낸다. 도 15에서, 노트북 패널은 세가지 탭, 즉 멤버 탭(1514), 서브 그룹

탭(1516), 그리고 애플렛 허가 탭(1518)을 포함하는 오른쪽 패널에 디스플레이된다. 멤버 탭은 도 19에서 선택된다. 멤버 패널은 시스템으로 정의된 모든 멤버의 로그-온 확인의 리스트(1520)를 포함한다. 새로운 유저를 생성하기 위해(현재 선택된 그룹 상황으로 멤버십을 자동적으로 얻는 "유저 그룹"), 관리자는 리스트(1520)로부터 <NE>를 선택하고, 엔트리 필드(1522) 안에서 리스트의 오른쪽으로 적절한 정보를 입력하고, 생성 버튼(1522)에서 클릭한다. 존재하는 멤버가 리스트(1520)로부터 선택될 때, 그 유저를 위해 이전에 세이브된 애트리뷰트가 (1522)에서 디스플레이된다. 이 애트리뷰트는 선택된 멤버의 완전한 이름, 멤버의 시스템 ID, 패스워드 그리고 어떠한 요구된 코멘트를 포함한다. ID를 제외한 애트리뷰트는 변경 버튼(1524)을 클릭함으로써 편집되고 변화가 실행되며, 또는 삭제 버튼(1526)을 클릭함으로써 유저는 시스템으로부터 완전히 제거된다. 어떠한 보류 변화도, 리스트(1520)안의 엔트리를 선택하고 Undo 버튼(1528)을 클릭함으로써 제거될 수 있다.

도 16은, 서브 그룹 탭(1516)이 선택될 때, 디스플레이되는 관리자의 오른쪽 패널을 나타낸다. 서브 그룹 리스트(1620)는 왼쪽 패널에서 선택되는 마이템의 서브 그룹인 존재하는 그룹을 나타내고, 이 그룹은 이 예에서 "유저 그룹"("User Group")이다. 그러므로, 리스트(1620)는 "AllUsers" 그룹의 즉각적인 서브 그룹을 디스플레이한다. 왼쪽 패널에서, "유저 그룹"은 확장된다. 리스트(1620) 안에 보여진 서브 그룹은 또한 왼쪽 패널 안의 "유저 그룹" 하의 확장된 아이tem이다. 리스트(1620)에서, 상태 필드는 "삭제"("Delete"), "변경"("Modify"), 그리고 "생성"("Create")과 같은 각 서브 그룹의 현재의 상태를 나타낸다. 리스트(1620) 안에서 빈 상태 필드는 서브 그룹이 존재하고 행동이 세이브되기 위해 보류되지 않음을 지시한다. 기호 "I"는 상태가 보류임(아직 세이브되지 않음)을 지시한다. 리스트(1620) 안에서 선택된 서브 그룹을 위한 애트리뷰트는 (1622) 안에서 나타난다. 이들 애트리뷰트는 서브 그룹 이름 및 서브 그룹에 관한 요구되는 코멘트를 포함한다. 새로운 서브 그룹을 생성하기 위해, 관리자는 리스트(1620)로부터 <NE>를 선택하고, (1622)에서 서브 그룹 이름 및 요구되는 코멘트를 입력하고, 생성 버튼(1628)을 클릭한다. "I 생성<서브 그룹 이름>"("Icreate<subgroupname>")의 엔트리는 보류 행위로서 리스트(1620) 안에 나타난다. 모든 보류 변경을 세이브하기 위해, 탭 메뉴 바(top menu bar) 그리고 SAVE(나타내어지지 않음)에서 파일 버튼을 클릭한다.

도 17은, 애플렛 허가 탭(1518)이 선택될 때, 디스플레이되는 오른쪽 패널을 보여준다. 리스트(1720)는 시스템에 정의된 모든 애플렛의 모든 이름을 나타내고, 그리고 왼쪽 패널에서 선택된 그룹 또는 서브 그룹(현재의 "상황")을 위한 각 애플렛에 지정된 허가 상태(허가 또는 접속 거절)를 나타낸다. 다른 노트 북 페이지가 설명된 바와 같이, 감탄 부호는 도시된 상태가 SAVE를 보류하는 변화임을 지시한다. 도 17에서, 그룹 "유저 그룹"은 왼쪽 패널 안에서 보여진 트리 안에서 선택되고, 이것은 도 3 안에서 보여진 "AllUsers" 그룹과 대응한다. 시스템의 모든 유저가 "유저 그룹" 그룹 안에서 멤버십을 가지고 있기 때문에, 리스트(1720)는 시스템으로 정의된 각각의 애플렛을 위한 모든 시스템 유저를 위해 전체적인 디플렛 허가 보여준다. 예를 들면, 애플렛 "데이터베이스 익스플로러"를 위한 디플렛 허가 상태는 "AllUsers" 그룹을 위한 "허가"("permit")이고(접속이 허가됨을 의미); 유사하게, 애플렛 TFTP로의 모든 유저를 위한 디플렛 허가 상태는 "거절"("deny")이다(접속이 거절됨). 관리자는 리스트(1720)에서 그것을 선택함으로써 그리고 "허가 접속 그룹"("Permit group access") 버튼(1730) 또는 "거절 그룹 접속"("Deny group access") 버튼(1732)을 클릭함으로써 애플렛의 허가 상태를 변경할 수 있다. 더욱이, 선택된 상황을 위한 애플렛의 허가 상태와 관계없이, 관리자는 선택된 상황하에서 유저 애플렛을 수행하기 위하여 (1720)로부터 애플렛을 선택하고 "작동/맞춤 제작" 버튼(1734)을 클릭할 수 있다. 현재의 상황을 위해 이전에 노트 북을 보여주는 패널 구역은 실행하는 유저 애플렛에 의하여 점유된다. 유저 애플렛이 다른 소프트웨어를 위한 구성 애플렛이라면, 관리자는, 선택된 상황을 위해 소프트웨어의 디플렛 프리퍼런스로서 세이브된 소프트웨어 프리퍼런스(이 기능을 위해 제공되는 구성 애플렛을 통해 독특한 파일)를 세이브할 수 있다. 애플렛이 엔드 유저 애플렛이라면, 엔드 유저 애플렛이 소프트웨어의 분리된 부분을 위한 프리퍼런스보다도 그 자신의 프리퍼런스를 로드하고 세이브하는 것을 제외하고는, 기능들은 동일하다.

도 18은 "유저 그룹" 아래에 관리자 왼쪽 패널 서브 그룹 트리의 완전한 확장을 보여준다. "유저 그룹" 바로 아래에, 2개의 서브 그룹 이 있으며, 제거될 수 없는 디플렛 서브 그룹인 "관리자"("Administrators"), 관리자에 의해 정의되는 서브 그룹인 "IBM"이다. "IBM" 서브 그룹은 또한 확장되고 그리고 세 개의 서브 그룹, "하드웨어", "서비스", 그리고 "소프트웨어"를 포함한다. "소프트웨어" 서브 그룹은 확장되고, 적어도 한 개의 서브 그룹, "디벨로퍼먼트"("Development")를 포함한다. "디벨로퍼먼트" 서브 그룹은 적어도 한 개의 서브 그룹, "NCod"를 포함한다. 서브 그룹 "NCod"는 힐드런(children)이 없는 ConfigFWS8과 같은 다수의 서브 그룹을 포함한다. 또한 이 예에서, 서브 그룹 "디벨로퍼먼트"는 확장 트리에서 선택된다. "디벨로퍼먼트"가 트리 계층 구조의 최상부에 있지 않으므로("AllUsers" 그룹), 모든 유저들이 "유저 그룹"을 가질 때, 모든 유저가 자동적으로 "디벨로퍼먼트"의 멤버는 아니기 때문에, "유저 그룹"이 선택될 때, 오른쪽 패널에 보여진 노트 북은 도 15의 것으로부터는 어느 정도 다르다. 리스트(1820)는 모든 시스템 멤버의 로그-온 시스템 ID를 디스플레이한다. 리스트(1820) 안에서 각 유저 ID 외의 상태는 유저가 "디벨로퍼먼트" 서브 그룹 안에서 멤버십을 소유하는지를 나타낸다. "yes"의 상태는 유저가 "디벨로퍼먼트" 서브 그룹의 멤버임을 지시하고, "no"는 유저가 "디벨로퍼먼트" 서브 그룹의 멤버가 아님을 지시하고, 그리고 "Inherited"는 유저가 트리 아래에 적어도 디벨로퍼먼트의 서브 그룹의 하나에 속함으로써 "디벨로퍼먼트" 그룹 내에 멤버십을 인계 받는 것을 지시한다. 서브 그룹을 위한 유저의 멤버십 상태는 리스트(1820) 안에서 유저를 선택하고 "Add to Group" 버튼(1836) 또는 "Remove from Group" 버튼(1838)을 클릭함으로써 관리자에 의해 변경된다. 관리자가 시스템 유저를 생성하거나, 또는 존재하는 멤버를 변경하거나 삭제하기를 원한다면, 관리자는 "생성/변경/삭제 유저" 버튼(1840)을 클릭한다. 이 행위는 도 19 안에 보여진 노트 북 페이지를 야기한다. 도 19의 오른쪽 패널은 도 15의 그것과 유사하며, 관리자가 리스트(1920) 안에서 NE를 선택하고 "생성" 버튼을 클릭함으로써 새로운 시스템 유저를 생성하는 것을 허용한다. 유사하게, 관리자는 리스트(1920) 안에서 적절한 유저를 선택하고 적절한 버튼 "변경" 또는 "삭제"를 클릭함으로써 존재하는 시스템 유저를 변경 또는 삭제할 수 있다. 어떠한 서브 그룹 상황(예를 들면, "디벨로퍼먼트")에서 생성된 유저가 "유저 그룹" 안에서 요구되는 멤버십을 얻을 뿐 아니라, 선택된 서브 그룹의 멤버로 자동적으로 만들어진다. 시스템 유저 리스트로의 변경은 오른쪽 패널의 탭 메뉴 바 안에서 "파일"을 클릭하고 다음으로 "세이브"를 클릭한

으로써 (나타내어지고 있지 않음) 세이브된다.

도 20은, 도 19 안에서 보여진 그룹 및 서버 그룹을 통하기보다 편집을 위한 시스템 유저 리스트에 도달하는 직접적인 방법을 보여준다. 도 20을 학습하기 위해, 관리자는 예를 들면 도 13의 왼쪽 패널 안의 "유저" (1304)를 선택한다. 도 20 안에서 보여진 오른쪽 패널에서, 관리자는, 그룹 또는 서버 그룹에 관계없이, 이미 전송된 바와 같이, 새로운 유저를 생성하고 존재하는 유저를 변경하고 삭제할 수 있다.

도 21에서, 관리자는, ID가 "콜랜드" ("collend")인 유저와 대응하는 정보 상에 직접적으로 작업하기를 원한다. 이것을 하기 위해, 관리자는, 예를 들면, 도 21의 왼쪽 패널 안에서 "유저"를 확장하고, 그리고 보여지는 바와 같이 "콜랜드"를 선택한다. 오른쪽 패널이 나타나고, 이것은 콜랜드의 시스템 정보에 종속한다. 오른쪽 패널은 세 가지 탭을 포함한다. 제 1 탭 "유저 정보"는 디폴트로 선택된다. 이 탭에서, 관리자는 콜랜드에 부속되는 이름, ID, 패스워드, 그리고 코멘트를 변경할 수 있다.

도 22는, 관리자가 제 2 탭 "그룹 멤버십"을 선택할 때, 오른쪽 패널을 보여준다. 리스트(2220)는 콜랜드가 멤버인 모든 서버 그룹을 보여준다. 서버 그룹은 콜랜드를 위해, 서버 그룹 우선 순위(에 따라 이 리스트 안에서 나타내어진다. 관리자는, 서버 그룹을 선택하고, 리스트(2220)의 오른쪽에 있는 위, 아래 화살표를 사용하여 선택된 서버 그룹을 변경하는 대로 리스트의 위 또는 아래로 움직이게 함으로써 콜랜드의 서버 그룹 우선 순위를 변경할 수 있다. 관리자가 도 22의 "Add/Remove Group Membership" 버튼(2242)을 클릭하면, 오른쪽 패널은 도 23의 콘텐츠를 보여준다. 도 23의 오른쪽 패널은, 관리자가 콜랜드가 멤버인 서버 그룹을 변경하는 것을 허용한다. 관리자는 요구되는 서버 그룹에 해당하는 적절한 박스(box)를 클릭함으로써 이것을 수행한다. 박스가 깨끗하다면(콜랜드가 현재 멤버가 아님을 의미), 검사 표시가 서버 그룹 안의 콜랜드를 포함하기 위해 박스로 더해진다. 역으로, 서버 그룹 박스가 이미 검사 되었으면, 박스를 클릭하면 검사 표시를 지우고, 서버 그룹으로부터 콜랜드를 제거한다.

도 24는, 도 22의 애플렛 허가 탭이 관리자에 의해 선택될 때, 오른쪽 패널을 보여준다. 이 오른쪽 패널에서, 리스트(2420)는 시스템 안에서 정의된 모든 애플렛을 디스플레이한다. 관리자는, 리스트(2420) 안에서 애플렛을 선택하고 "허가 유저 접속" 버튼(2430)을 클릭함으로써 콜랜드에 의해 애플렛으로 접속을 허가할 수 있고, 또는 접속이 거절, 유저 접속" 버튼(2432)을 클릭함으로써 콜랜드로 거절할 수 있다. 관리자는 "작동/맞춤 제작" 버튼(2434)을 클릭함으로써 콜랜드의 관계에서 또한 애플렛을 학습할 수 있다. 이것이 이루어질 때, 리스트(2420) 안에서 선택된 애플렛은 오른쪽 패널에서 학습된다. 관리자는 애플렛이 허용하는 어떠한 프레퍼런스를 변경할 수 있고 애플렛에 의해 제공되는 방법으로 프레퍼런스를 세이브할 수 있다. 여기서 전형적인 시나리오는 관리자가 여러 형태의 프레퍼런스 필드를 채우기 위해 구성 애플렛을 학습하는 것이다. 그러나, 분리된 구성이 유저 애플렛을 위해 제공되지 않는다면, 관리자가 유저와 관계하여 유저 애플렛을 학습할 수 있고 유저 애플렛으로부터 프레퍼런스를 정할 수 있다. 여기서 전형적인 시나리오는 관리자가 그룹 또는 유저 상황을 선택하고, 전송된 바와 같이 유저 애플렛을 학습하는 것이다. 관리자는 옵션 메뉴로부터 프레퍼런스를 전형적으로 변경할 수 있고, 유저 애플렛에 의해 제공되는 방법으로 그들을 저장할 수 있다. 예를 들면, 전형적으로, 유저 프레퍼런스는 옵션 다이얼로그가 폐쇄될 때 저장되고, 또는 유저 애플렛은 프레퍼런스를 저장하는 다른 방법을 제공할 수 있다. 어떠한 이벤트에서, 이 예에서, 관리자가 콜랜드와 관계되어 애플렛을 작동하므로, 콜랜드가 애플렛을 작동함으로써 프레퍼런스를 입력시킨 것처럼, 유저 애플렛을 통해 관리자에 의해 셋업되는 프레퍼런스를 이 서버 상에서 세이브된다.

유저가 유저 애플렛에 속하는 일부 프레퍼런스를 변경할 수 있는 시나리오는 도면에 나타나지 않는다. 예를 들면, 유저 애플렛은 유저가 윈도우 배경색 또는 폰트 그리고 폰트 크기를 선택하는 것을 허용하고, 따라서 각 시스템은 유저 애플렛이 유저 데스크탑 상에서 실행할 때 어느 정도로 애플렛을 개별화 할 수 있다. 이 경우에, 유저 변경된 프레퍼런스는, 관리자가 유저 애플렛을 작동할 때, 그들이 세이브된 것처럼 동일 방법으로 세이브된다. 그러나, 하나의 차이점은 유저들이 그들의 개별 상황을 위해 오직 프레퍼런스에 영향을 줄 수 있는 반면에, 관리자가 그룹 상황 안에서 프레퍼런스를 정하기 위해 유저 애플렛을 작동시킬 수 있다.

전송된 장치가 단지 본 발명의 원리의 적용을 설명적으로 나타내고, 다른 장치들이 본 발명의 사상과 범주를 벗어나지 않고 이 분야의 숙련자들에 의해 고안될 수 있음이 이해되어야 한다.

발명의 효과

본 발명의 원리에 따르면, 관리자와 유저간의 네트워킹(networking) 환경에서, 유저(user), 또는 유저 그룹(user group)의 관계에서 직접적으로 애플리케이션(application)을 작동시킴으로써 관리자가 유저 애플리케이션을 구성(configuration)하는 것이 가능하며, 즉 애플리케이션의 구성이, 애플리케이션을 실행하고 이에 의해 제공되는 옵션을 실행하고, 그리고 구성을 저장함으로써 수행되는 미점을 얻을 수 있다.

(57) 청구의 범위

청구항 1

서버(server)와 복수의 유저 스테이션(user station)을 상호 연결하는 네트워크(network)를 포함하는 네트워크 시스템에서, 서버가 유저 스테이션으로 다운로드(downloading)하기 위해 복수의 유저 애플리케이션(application)을 저장하고, 유저 스테이션에서 실행하는 애플리케이션을 위한 유저 구성(configuration) 프레퍼런스(preference)를 관리하는 방법에 있어서,

관리자 스테이션에서 프로필 매니저(profile manager)를 제공하는 단계;

제 1 엔드 유저 애플리케이션(end user application)을 위해 분리된 구성 애플리케이션을 실행하도록 상기 프로필 매니저를 배열하는 단계 - 그함으로써, 관리자는 시스템 유저의 다른 그룹 및 서버 그룹의 관

제에서 상기 제 1 엔드 유저 애플리케이션을 위해 구성 프레퍼런스를 특정할 수 있음 - ;

상기 서버에서 상기 제 1 엔드 유저 애플리케이션을 위한 상기 구성 프레퍼런스를 저장하는 단계 ;

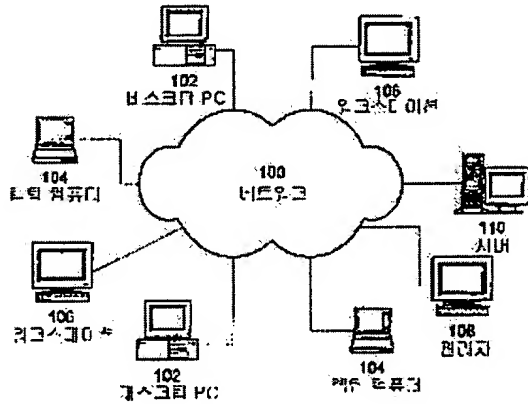
상기 관리자의 스테이션에서 제 2 엔드 유저 애플리케이션을 실행함으로써 다른 그룹 및 서브 그룹의 관계에서 상기 제 2 엔드 유저 애플리케이션을 위해 구성 프레퍼런스를 특정할 목적으로 유저들의 다른 그룹 및 서브 그룹의 관계에서 제 2 엔드 유저 애플리케이션을 실행하기 위해 상기 프로필 매니저를 배열하는 단계 ; 및

상기 서버 상에서 상기 제 2 엔드 유저 애플리케이션을 위한 상기 구성 프레퍼런스를 저장하는 단계를 포함하는

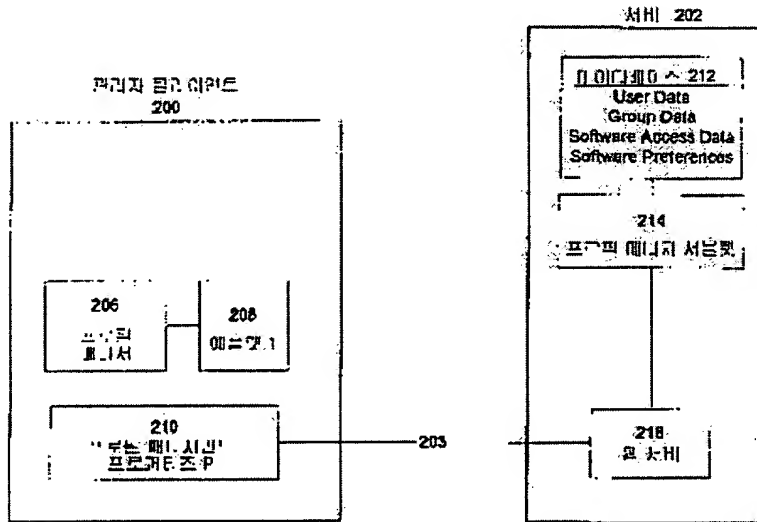
유저 구성 프레퍼런스를 관리하는 방법 .

도면

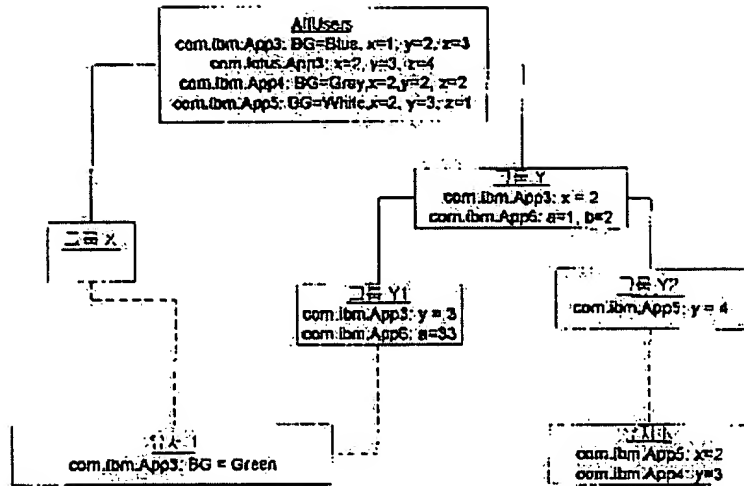
도면1



도면2



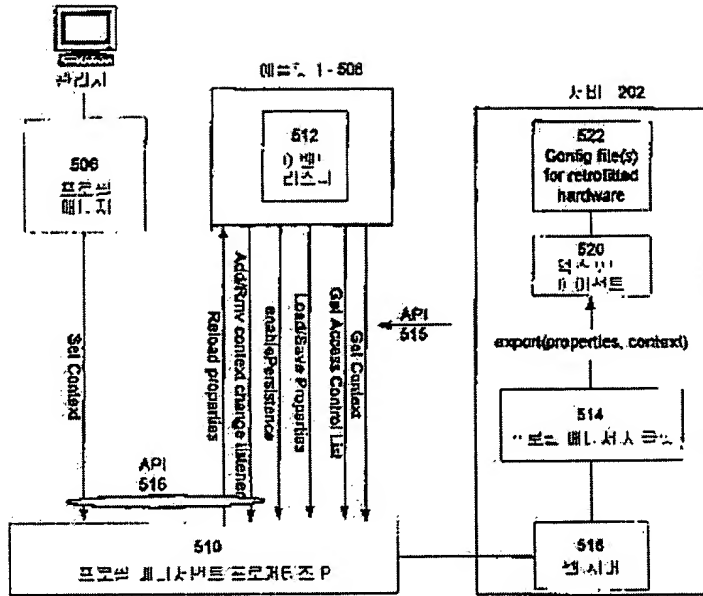
CB3



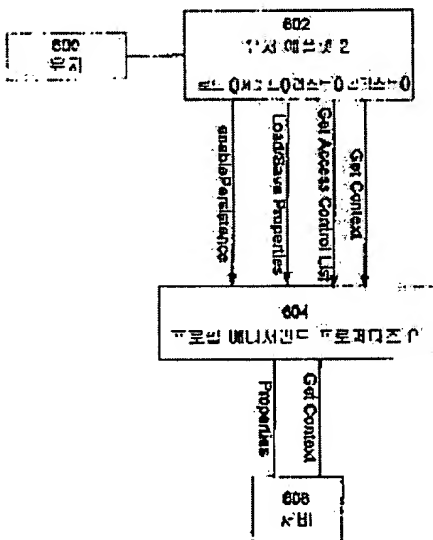
CB4

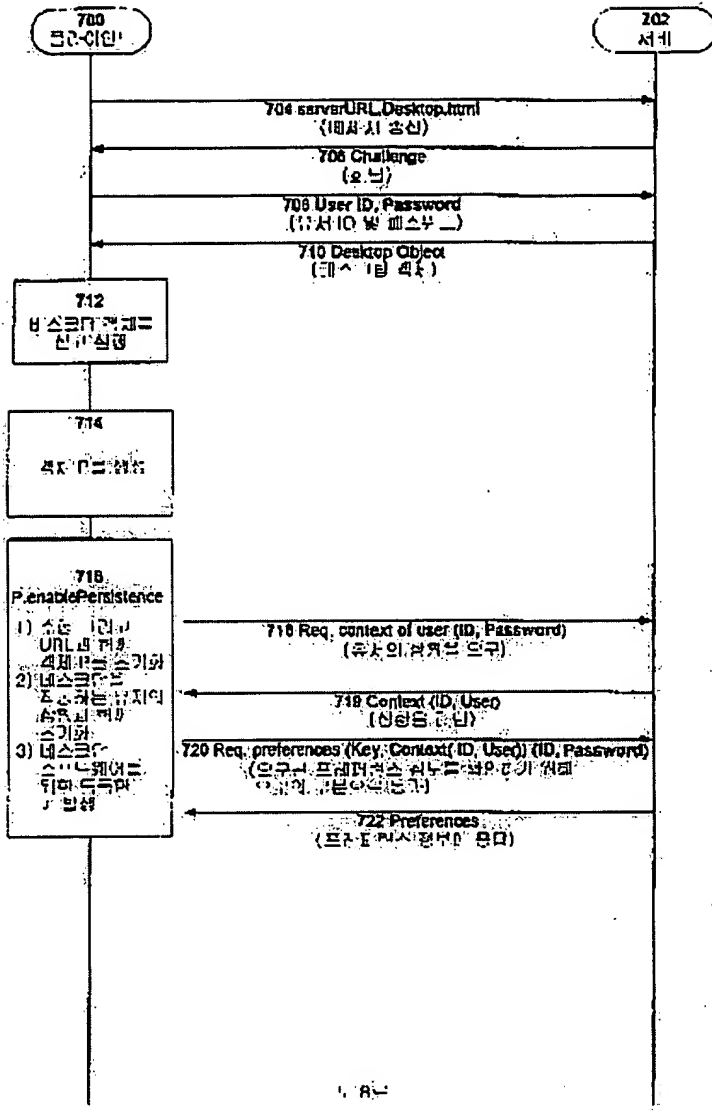
구분	이름	일련번호
User1	AllUsers.GroupX	1
	AllUsers.GroupY.GroupY1	2
	AllUsers	3
UserN	AllUsers.GroupY.GroupY2	1
	AllUsers	2

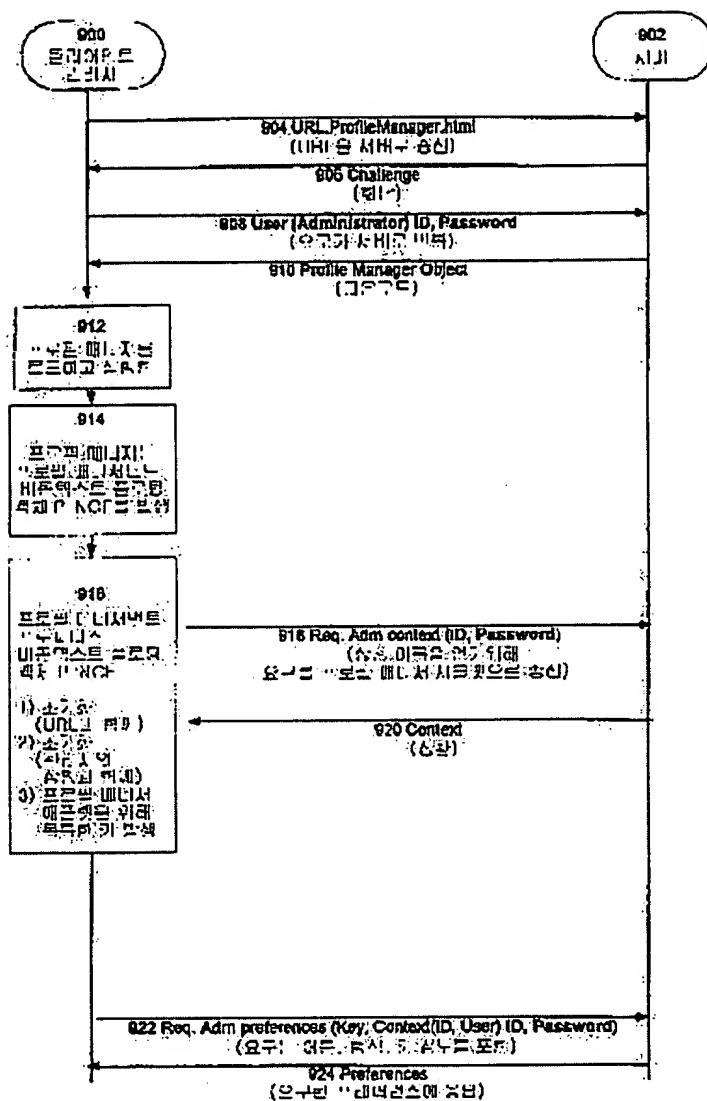
도 5



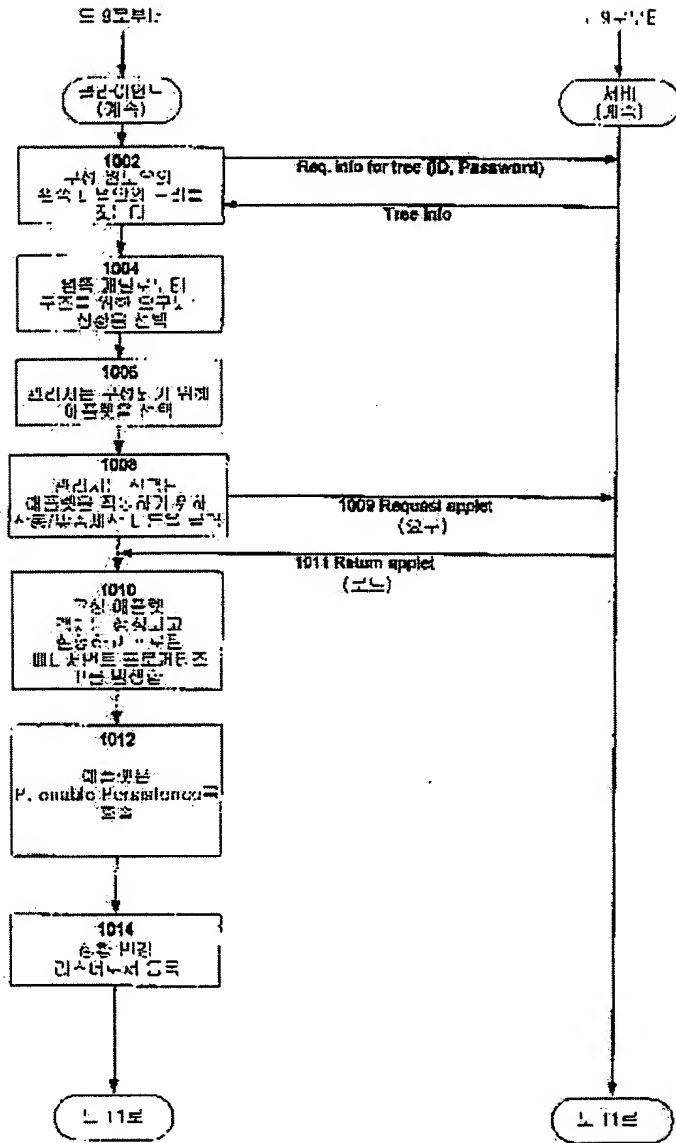
도 6



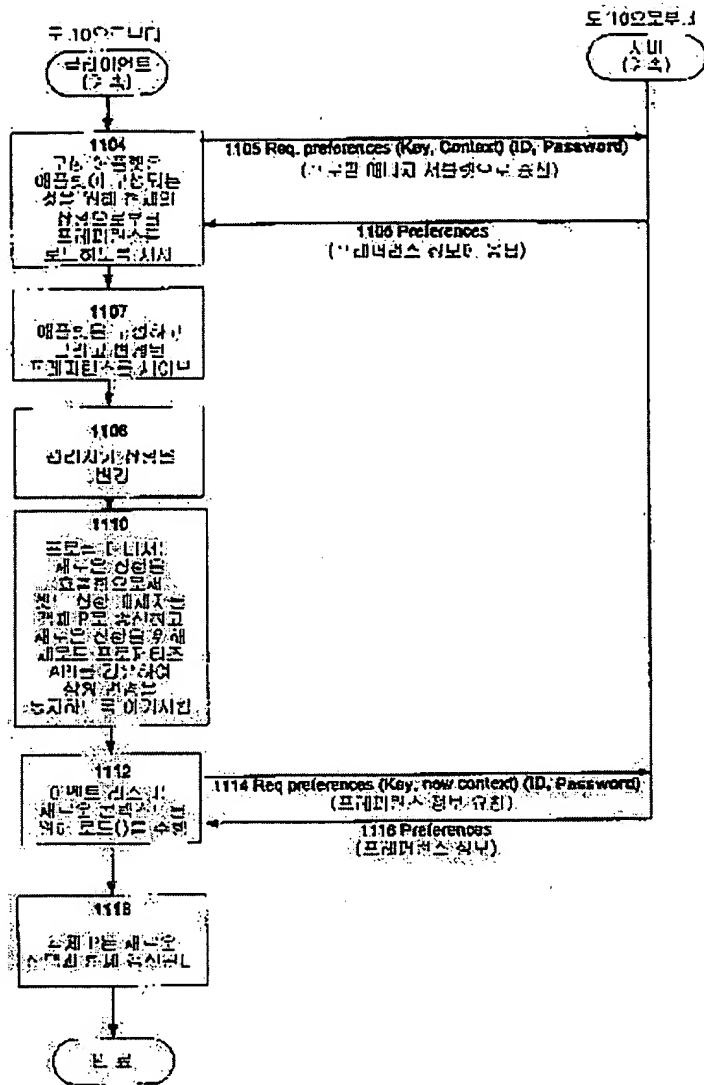


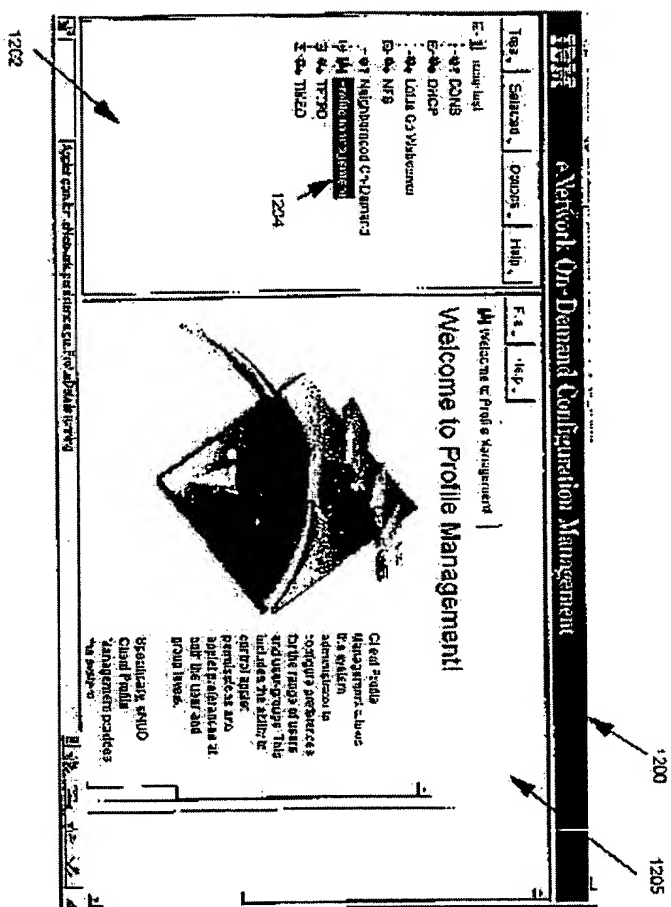


도면 10



도 11





34-23

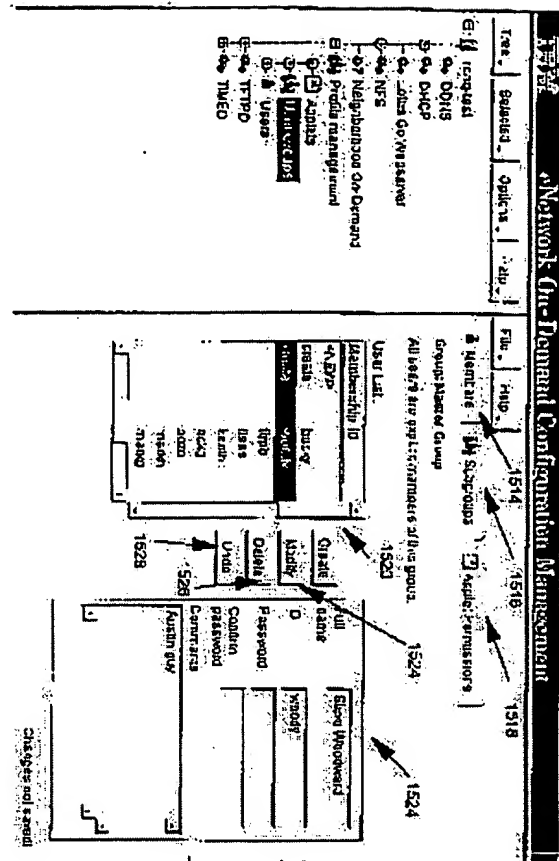
5014

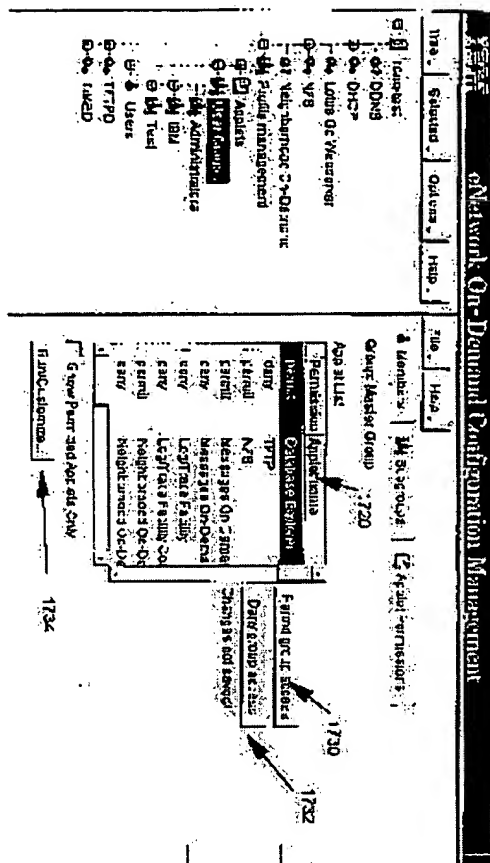
Import Applied List From File
Note: If an Applied List from a .c will cause any pending changes to be lost
then save it first.

Complete any of the

IR: 2008-01-01

2008-01-01





eNetwork: On-Demand Configuration Management

[illegible]

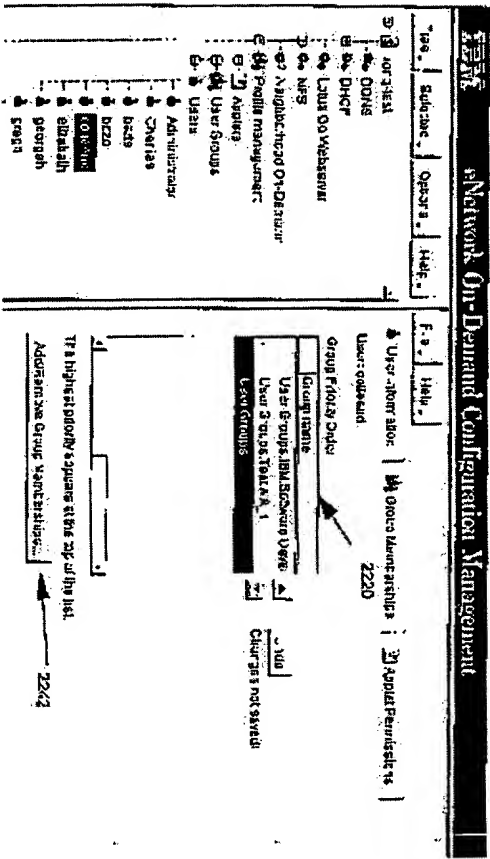
Time	Release	Category	File
01	0000-0001	0000	0000
02	0000-0001	0000	0000
03	0000-0001	0000	0000
04	0000-0001	0000	0000
05	0000-0001	0000	0000
06	0000-0001	0000	0000
07	0000-0001	0000	0000
08	0000-0001	0000	0000
09	0000-0001	0000	0000
10	0000-0001	0000	0000
11	0000-0001	0000	0000
12	0000-0001	0000	0000
13	0000-0001	0000	0000
14	0000-0001	0000	0000
15	0000-0001	0000	0000
16	0000-0001	0000	0000
17	0000-0001	0000	0000
18	0000-0001	0000	0000
19	0000-0001	0000	0000
20	0000-0001	0000	0000
21	0000-0001	0000	0000
22	0000-0001	0000	0000
23	0000-0001	0000	0000
24	0000-0001	0000	0000
25	0000-0001	0000	0000
26	0000-0001	0000	0000
27	0000-0001	0000	0000
28	0000-0001	0000	0000
29	0000-0001	0000	0000
30	0000-0001	0000	0000
31	0000-0001	0000	0000
32	0000-0001	0000	0000
33	0000-0001	0000	0000
34	0000-0001	0000	0000
35	0000-0001	0000	0000
36	0000-0001	0000	0000
37	0000-0001	0000	0000
38	0000-0001	0000	0000
39	0000-0001	0000	0000
40	0000-0001	0000	0000
41	0000-0001	0000	0000
42	0000-0001	0000	0000
43	0000-0001	0000	0000
44	0000-0001	0000	0000
45	0000-0001	0000	0000
46	0000-0001	0000	0000
47	0000-0001	0000	0000
48	0000-0001	0000	0000
49	0000-0001	0000	0000
50	0000-0001	0000	0000
51	0000-0001	0000	0000
52	0000-0001	0000	0000
53	0000-0001	0000	0000
54	0000-0001	0000	0000
55	0000-0001	0000	0000
56	0000-0001	0000	0000
57	0000-0001	0000	0000
58	0000-0001	0000	0000
59	0000-0001	0000	0000
60	0000-0001	0000	0000
61	0000-0001	0000	0000
62	0000-0001	0000	0000
63	0000-0001	0000	0000
64	0000-0001	0000	0000
65	0000-0001	0000	0000
66	0000-0001	0000	0000
67	0000-0001	0000	0000
68	0000-0001	0000	0000
69	0000-0001	0000	0000
70	0000-0001	0000	0000
71	0000-0001	0000	0000
72	0000-0001	0000	0000
73	0000-0001	0000	0000
74	0000-0001	0000	0000
75	0000-0001	0000	0000
76	0000-0001	0000	0000
77	0000-0001	0000	0000
78	0000-0001	0000	0000
79	0000-0001	0000	0000
80	0000-0001	0000	0000
81	0000-		

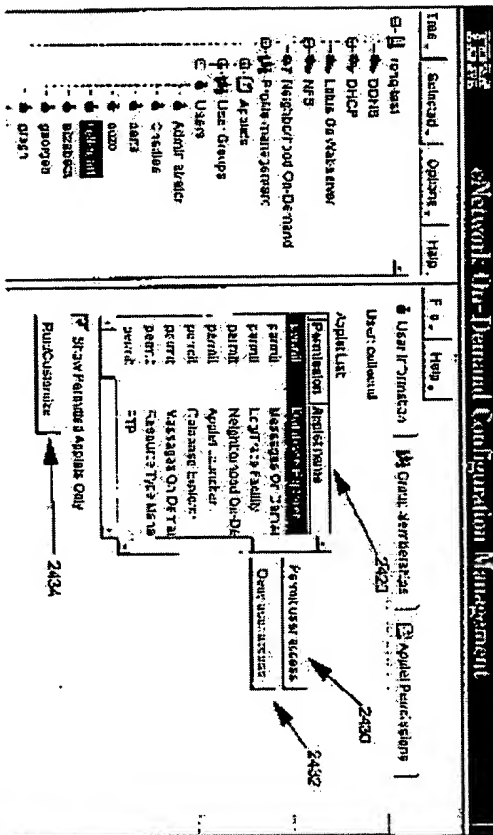
The screenshot displays the 'Network On-Demand Configuration Manager' application. The interface includes a standard menu bar (File, Edit, View, Options, Help) and a status bar at the bottom. The main workspace is organized into three primary sections:

- User List:** A table listing network users.

ID	Name	Address
1	dell	192.168.1.100
- Create:** A section for adding new users, featuring input fields for 'Name', 'Address', and 'IP Address'.
- Full:** A detailed configuration area containing:
 - A table listing network components:

ID	Name	Address	Password
1	Router	192.168.1.1	12345678
 - A 'Connections' section with a 'New Connection' button and a list of existing connections.





**This Page is Inserted by IFW Indexing and Scanning
Operations and is not part of the Official Record**

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

☒ **BLACK BORDERS**

☐ **IMAGE CUT OFF AT TOP, BOTTOM OR SIDES**

☐ **FADED TEXT OR DRAWING**

☐ **BLURRED OR ILLEGIBLE TEXT OR DRAWING**

☐ **SKEWED/SLANTED IMAGES**

☒ **COLOR OR BLACK AND WHITE PHOTOGRAPHS**

☒ **GRAY SCALE DOCUMENTS**

☐ **LINES OR MARKS ON ORIGINAL DOCUMENT**

☐ **REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY**

☐ **OTHER:** _____

IMAGES ARE BEST AVAILABLE COPY.

As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.

THIS PAGE BLANK (USPTO)